Registration of Expressions Data using a 3D Morphable Model

Curzio Basso, Pascal Paysan, Thomas Vetter
Computer Science Department, University of Basel
{*curzio.basso,pascal.paysan,thomas.vetter*}*@unibas.ch*

Abstract

The registration of 3D scans of faces is a key step for many applications, in particular for building 3D Morphable Models. Although a number of algorithms are already available for registering data with neutral expression, the registration of scans with arbitrary expressions is typically performed under the assumption of a known, fixed identity. We present a novel algorithm which breaks this restriction, allowing to register 3D scans of faces with arbitrary identity and expression. Furthermore, our algorithm can process incomplete data, yielding results which are both continuous and with low reconstruction error. Even in the case of complete, expression-less data, our method can yield better results than previous algorithms, due to an adaptive smoothing, which regularizes the results surface only where the estimated correspondence is unreliable.

1. Introduction

The registration of 3D scans of human faces is a key step in their processing for many applications. We present an algorithm closely related to the methods using a regularized energy minimization ([15, 11, 1]). This is a common approach, since the regularization term provides the advantage of handling missing data and inconsistencies in the *correspondence*. The correspondence can be derived from a manually-defined sparse correspondence ([13, 10]), or with ICP ([1]). We estimate a dense correspondence following an approach similar to [6, 3]. This results in a more accurate registration, and the correspondence is also used to determine the local importance of the regularization term. Our registration algorithm presents three novel characteristics.

Unified Processing. Although some very efficient methods for registering 3D scans of human faces have already been published ([6, 11, 10, 4, 19]), the registration of data with varying identities and expressions are typically treated separately. An exception is the method in [18], which however needs around 70 user-defined landmark points. Our algorithm can be applied to 3D face scans with arbitrary identity and expressions, which makes it suitable for the applications where no such prior knowledge is available (e.g. recognition).

Reconstruction of Missing Data. The input data of the registration algorithm is typically incomplete. In previous methods this problem is either not considered or it is addressed from a purely geometric point of view, a clear drawback if the results of the registration have to be used to build statistical models of human faces. In our registration algorithm the reconstruction of the missing areas takes into account not only its geometric properties but also its likelihood w.r.t. already available data.

Robustness. A further novelty of our algorithm is related to the estimation of *correspondence*. In methods which regularize the result ([15, 11, 1]), the relative importance of the correspondence in the registration process – for instance with respect to the smoothness of the registration result – is globally fixed. This might either result in loss of information, or introduce errors in the registration results. By setting the relative importance of the correspondence locally, our algorithm retains as much correspondence information as possible, while at the same time being robust with respect to errors in its estimation.

2. 3D Morphable Models

Before describing the registration algorithm, we review the concept of the 3D Morphable Model (3DMM) and show how the 3DMM is extended to handle both identity and expression as separate sources of variations. The shape of a 3D mesh with n vertices is represented as an $n \times 3$ matrix S, or alternatively as a $3n$-dimensional column vector obtained by flattening the matrix:

$$s = vec(S) = (x_1, y_1, z_1, \ldots, x_n, y_n, z_n)^T. \qquad (1)$$

A similar representation can be used for the texture of the 3D mesh, which is stored in a vector t:

$$t = vec(T) = (r_1, g_1, b_1, \ldots, r_n, g_n, b_n)^T. \qquad (2)$$

In this section we will develop the model of the shape, using the vector representation. The texture model is obtained following the same procedure.

Three-dimensional morphable models are built under the assumption that a shape vector s is generated by a *linear Gaussian model*, defined by a mean vector $\overline{s} \in \mathbb{R}^{3n}$ and a generative matrix $C \in \mathbb{R}^{3n \times k}$ with $k < 3n$.:

$$s = \overline{s} + C \cdot \alpha + \varepsilon. \tag{3}$$

The vectors $\alpha \in \mathbb{R}^k$ and $\varepsilon \in \mathbb{R}^{3n}$ are the latent variables of the model, and they follow a Gaussian distribution with zero mean and diagonal covariance:

$$p(\alpha) = \mathcal{N}(0, I) \quad \text{and} \quad p(\varepsilon) = \mathcal{N}(0, \sigma^2 I). \tag{4}$$

The model parameters $\overline{s}$, C and σ^2 can be estimated by maximizing the likelihood of a training set of examples shapes $s_1, \ldots, s_m$. We report here only the solution of the maximization, and refer to [16] for details. Defining $\overline{s}$ as the sample mean

$$\overline{s} = \frac{1}{m} \sum_{i=1}^{m} s_i, \tag{5}$$

we decompose the *centered* data matrix by *Singular Value Decomposition* (SVD):

$$\begin{aligned} A &= (s_1 - \overline{s}, \ldots, s_m - \overline{s}) \in \mathbb{R}^{3n \times m} && (6) \\ &= U \cdot W \cdot V^T. && (7) \end{aligned}$$

Recall that U is a column-orthogonal matrix ($U^T U = I$) and that W is a diagonal matrix with elements w_i. Denoting by Λ the diagonal matrix with elements $w_i^2/(m-1)$, the optimal estimates of C and σ^2 are given by

$$\begin{aligned} \sigma^2 &= \frac{1}{(m-1)(3n-k)} \sum_{i=k+1}^{m-1} w_i^2, && (8) \\ C &= U_k \cdot \left(\Lambda_k - \sigma^2 I\right)^{1/2}, && (9) \end{aligned}$$

where k is the number of principal directions which are retained, and the matrices U_k and Λ_k are obtained from the first k columns of U and Λ, respectively. The difference from this model and the one obtained from PCA (in the case of $k = m-1$) is that discarding some of the higher components u_i, their contributions to the sample variance accumulates in the model noise and scales down the variance of the retained components.

2.1. Combined Model

In order to model expressions and identity as separate sources of variations, we assume that a generic face shape is the sum of an identity vector and an expression vector:

$$s = s_{id} + s_{xp}, \tag{10}$$

while the face texture depends only on the identity. The vectors s_{id} and s_{xp} holds respectively the face shape with neutral expression and the displacements of the vertices due to the expression; assigning them separate linear Gaussian models we obtain:

$$s = \overline{s}_{id} + C_{id} \cdot \alpha_{id} + \overline{s}_{xp} + C_{xp} \cdot \alpha_{xp} + \varepsilon, \tag{11}$$

with the usual Gaussian prior for the latent variables α_{id}, α_{xp} and ε. Clearly, once the model parameters are fixed, this is equivalent to the model of eq. (3), with the only difference that the matrix $C = [C_{id}\ C_{xp}]$ is not column-orthogonal.

In order to learn the distinct model parameters for the identity and expressions components we use two training sets. A first set of examples with neutral expression and varying identity is used to estimate the identity parameters $\overline{s}_{id}$ and C_{id}, as outlined in the previous section. The expression parameters $\overline{s}_{xp}$ and C_{xp} are estimated from a second set of expression examples, acquired from p different persons. Given the i-th individual, we have its neutral expression n^i and m_i examples s^i_j, from which we build a matrix

$$B^i = (s^i_1 - n^i, \ldots, s^i_{m_i} - n^i) \in \mathbb{R}^{3n \times m_i}. \tag{12}$$

All the person-specific matrices B^i are then put together into a matrix

$$B = (B^1 \ldots B^p) \in \mathbb{R}^{3x \times \sum m_i}. \tag{13}$$

The average expression $\overline{s}_{xp}$ is computed as the mean of the columns of B, which is then recentered and decomposed by SVD to obtain the matrix C_{xp} as in eq. (9).

3. Registration Algorithm

The shape of a novel mesh is registered in three steps (see also the diagram of figure 1): first, the morphable model defined in the previous section is used to approximate the input mesh; then, the correspondence is estimated between the approximation and the input mesh; in the third step, the shape is registered by solving an optimization problem. For the registration of the texture, the third step is different; we will describe it in section 3.2, after having explained how the shape is processed.

Approximation. In order to obtain a more accurate estimation of the correspondence we employ a strategy known as *bootstrapping* (see [17]). Given a 3DMM, its coefficients are optimized with a stochastic Newton descent method so that its shape and texture fit the input

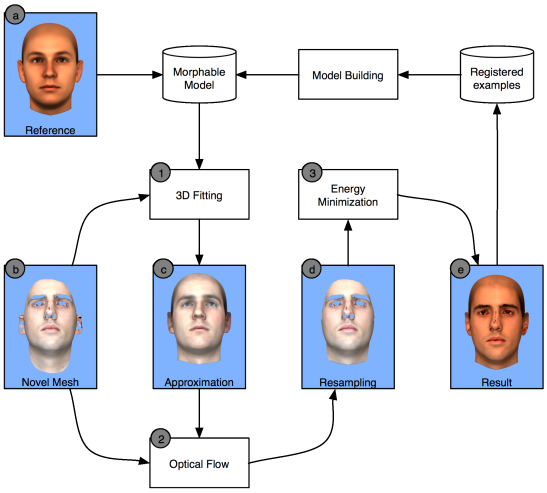

Figure 1. Flow diagram of the registration method. (1) The morphable model is fitted to the novel mesh (b). (2) A correspondence is estimated via optical flow between the approximation (c) and the input. Using the correspondence, the input is resampled yielding the incomplete surface (d). (3) The registration result (e) is obtained by minimizing an energy which depends on the resampling of the input. Each result of the registration increases the set of examples used to build the morphable model.

mesh (see [6]). An example is shown in figure 1(c); the novel mesh is in figure 1(b) and the reference in figure 1(a). In case no 3DMM is yet available, this step is skipped and a reference model is used as approximation.

Correspondence Estimation. Both the input mesh and the approximation are projected to a cylindrical 2D representation. We estimate a correspondence between them with the modified optical flow algorithm described in [6], using both the shape and texture information. The optical flow defines a correspondence between the vertices a_i of the approximation and points w_i lying on the surface of the input (see figure 2). We face now two problems: the input data are typically incomplete (see figure 1(d)), and the optical flow is not everywhere reliable. For these reasons, the vertices positions w_i computed through optical flow cannot be directly used to define the registration result.

Energy minimization. We compute the final results v_i minimizing an energy made up of a data term, depending on the positions w_i obtained from optical flow, and a smoothness term. This allows us to reconstruct the positions of the vertices without correspondence, and to regularize the positions of the vertices where the correspondence is unreliable.

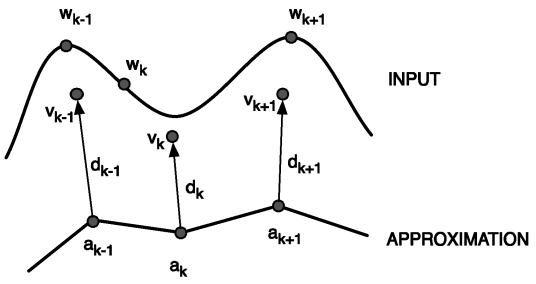

Figure 2. Notation used for defining the energy minimized during registration. The positions of the vertices in the approximation are denoted by a_i, the corresponding sampled positions on the input by w_i, and the unknown vertices positions in the solution by v_i. The displacements from the approximation to the solution are denoted by d_i.

3.1. Energy Minimization

Since the positions w_i are defined only for the vertices with a correspondence, we define the data term only for the subset $\mathscr{C}$ of such vertices:

$$E_d = \sum_{i \in \mathscr{C}} \|v_i - w_i\|^2 \tag{14}$$

Let a_i denote the positions of the vertices in the approximation; the regularization term depends on the displacements with respect to the approximation $d_i = v_i - a_i$ (see figure 2):

$$E_s = \sum_i \sum_{j \in \mathscr{N}_i} e_{ij} \|d_j - d_i\|^2, \tag{15}$$

where $\mathscr{N}_i$ denotes the neighborhood of the i-th vertex, and the coefficients e_{ij} weights the relative importance of each edge to the smoothness energy of a vertex. A good criterion for the choice of the coefficients e_{ij} is to look at how much the edges deform in the examples already registered. Defining with σ_{ij} the standard deviations of the edges lengths over the examples, we set

$$e_{ij} = \sigma_{ij}^{-2} / \sum_{\mathscr{N}_i} \sigma_{ij}^{-2}, \tag{16}$$

Substituting eq. (16) in eq. (15), we can verify that with this choice the smoothness term for each vertex becomes

$$\sum_{j \in \mathscr{N}_i} e_{ij} \|d_j - d_i\|^2 \propto \sum_{j \in \mathscr{N}_i} \|d_j - d_i\|^2 / \sigma_{ij}^2, \tag{17}$$

so that the influence of each edge on the vertex energy is weighted by its deformations in the available examples.

Figure 3. Reconstruction of texture coordinates for the eyes. In the top image we show the texture resulting from the push-pull algorithm in the eyes region. A much better result can be obtained by applying the push-pull algorithm on the texture coordinate data (bottom).

The adaptive smoothing is achieved by weighting each term of E_d with a coefficient λ_i:

$$E = \frac{1}{2}\sum_{i\in\mathscr{C}} \lambda_i \|v_i - w_i\|^2 + \frac{1}{2}\sum_i \sum_{j\in\mathscr{N}_i} e_{ij}\|d_j - d_i\|^2. \quad (18)$$

Note that the positions of the vertices without correspondence are determined only by the regularization term; its minimization produces a reconstruction of the missing vertices which is continuous and with low reconstruction error, as shown in [2]. The coefficients λ_i should be large where the correspondence is reliable, to let the data term dominate, and small otherwise, to regularize the result. As a measure of the correspondence quality, we use the smoothness of the displacement field $w_i - a_i$, defined as

$$s_i = \sum_{\mathscr{N}_i} \frac{\|(w_j - a_j) - (w_i - a_i)\|^2}{\|a_j - a_i\|^2}. \quad (19)$$

As shown in the example of figure 5, high values of this quantity are an index of problems in the correspondence. In our experiments, we set the λ_i to 10 for $s_i < 0.2$, to 10^{-7} for $s_i \geq 1$, and to 10^{-2} for $0.2 \leq s_i < 1$, which produces a slight smoothing. Of course more fine-grained choices of λ_i are possible, but in our experiments this choice proved to be sufficiently flexible.

The global minimum of the energy (18) is found by setting to zero its derivative w.r.t. the unknowns v_i. This yields a sparse linear system, which can be efficiently solved with standard algorithms (in our implementation we used [7]). Denoting by D, A and W the $N \times 3$ matrices holding the values of the vectors d_i, a_i and w_i, the system is

$$(\Lambda + I - (K + K^T)/2) \cdot D = -\Lambda \cdot (A - W), \quad (20)$$

where Λ is an $n \times n$ diagonal matrix with elements $\lambda_i/2$, and K is a sparse $n \times n$ matrix with $K_{ij} = e_{ij}$ if $\{i, j\}$ is an edge of the mesh and $K_{ij} = 0$ otherwise. Solving eq. (20) for D, the registered positions of the vertices are found as $v_i = a_i + d_i$.

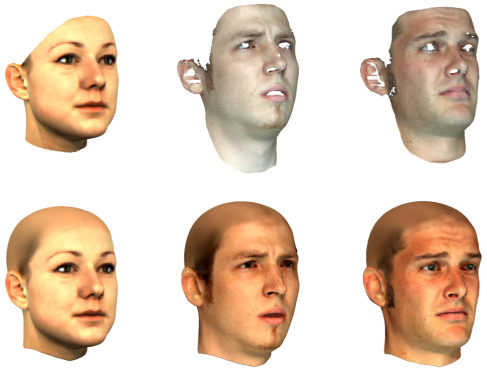

Figure 4. Three examples from our dataset, the originals on the top row and the registration results on the bottom row. Note how the texture in the last two examples is corrected during registration in order to be consistent with the other examples in the model.

3.2. Texture processing

With current 3D acquisition technologies, the 3D scans are typically texture mapped with high resolution images. Therefore, due to the different nature of the data, the texture registration is performed following a different procedure. Using the correspondence estimated by the optical flow, we assign to each vertex of the approximation a texture coordinate of the novel mesh. This allows to texture parts of the result directly with the original images, without any loss of information, but not the whole mesh. In order to obtain a complete texture, we warp the original texture to a fixed texture map, and then reconstruct the missing texture with a method based on the push-pull algorithm presented in [8]. The original algorithm diffuses the known color values to the missing regions, by iteratively down-sampling and up-sampling the image while keeping constant the known area. We apply the algorithm to the difference between the warping of the original texture and the approximation obtained during the first step of the registration. The result of the diffusion is then added to the approximating texture. It might also occur that holes in the acquired surface prevent the use

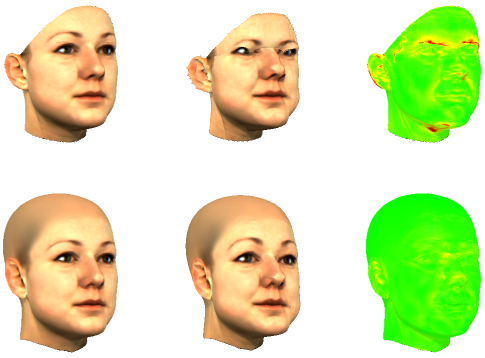

Figure 5. Registration of an example with our algorithm (bottom row) and the algorithm of [6] (top row). In the leftmost column are the registration results and in the middle column their shape caricatures. The caricature on top evidences the correspondence problems of the previous algorithm. On the rightmost column, a color-coded rendering (red is lower smoothness, green higher) of the displacement smoothness, shows that this measure can be used to detect correspondence problems.

of parts of the texture images, since the texture coordinates are not present if the surface is missing. In this case we apply the push-pull method described above to the texture coordinates before sampling them. In this way, we can use the original texture information also for areas where the surface could not be reconstructed (see figure 3).

4. Results

In order to test our algorithm, we applied it to an heterogeneous collection of 485 3D scans, acquired in part with a Cyberware scanner and in part with a phase shift system. The collection is equally divided in examples with neutral expression (233 scans, all with different identities) and examples with emotional expressions or visemes (252 scans, from 33 subjects). The whole dataset has been registered starting from two reference meshes of a full head with open and closed mouth. Figure 4 shows different examples of the training data and the results of their registration.

As we mentioned in the introduction, our algorithm is robust to errors in the correspondence estimation, thanks to the smoothness term in the minimized energy. Although there is no obvious way to measure its robustness, we assume that the smoothness of the displacement field between the registered results and their average shape is a reliable way to detect problematic areas

Figure 6. Average distance of the vertices from the original surface, ranging from 0.0 mm (green) to ≥ 1.0 mm (red). The black areas correspond to vertices missing in at least one example. Most of the vertices are close to the surface.

of the results, as shown in figure 5. A comparison between the average values of this smoothness for the results of our registration algorithm and the results of an algorithm based only on the correspondence estimation ([6]) confirms, as expected, that our method yields more regular results (the lower the better): 0.138 ($\sigma = 0.086$) vs. 0.204 ($\sigma = 0.182$). To rule out the possibility that the results are too smooth, resulting in a bad approximation of the input, we also checked the distances between the registered vertices and the input surfaces. The results, summarized in figure 6, show that the smoothing really affects the distance from the original surface only in small areas. On the rest of the face the vertices are within a distance of 1.0 mm from the surface.

We conclude this section by showing that the improvement in the quality of the results has also a positive effect on the quality of the morphable model.To this aim, we performed a 10-fold *cross-validation* (for more details see [9]) on two models built with the registration results of the previous comparison, in order to estimate the *generalization error* of the model. This is the expected error made by the model in reconstructing novel data, which was not in the training set. As shown in figure 7, the new results provide a model which is much more compact: 40 components are enough to achieve an error smaller than using 170 components of the old model.

5. Conclusion

We described an algorithm aimed at registering 3D face data with arbitrary identity and expression. The algorithm also allows us to register data with missing values and offers a control on the regularity of the registered results, thanks to the adaptive smoothing performed in the last step of the registration. In fact, the last step is a generalized version of the surface reconstruction method we described in [2], where we showed that its results are better than a purely statistical reconstruction (as in [5]).

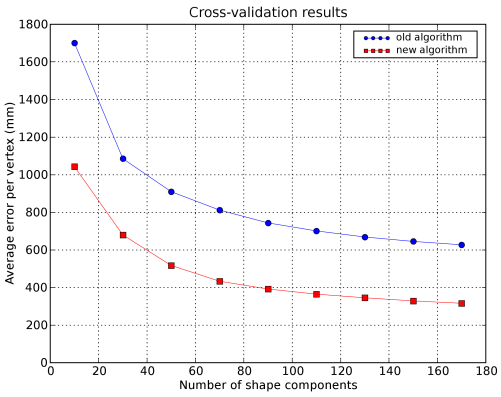

Figure 7. Generalization errors of the 3DMMs obtained with our algorithm and the algorithm of [6].

With the new algorithm we were able to build a 3DMM combining both identity and expression variations, which in principle might be used in many applications, as well as in the registration itself: face recognition (both in 3D and 2D), video tracking, face animation, expression normalization in images. Although we used a linear model, one could also use a *bilinear model* as in [18]. In principle such a model has the advantage of capturing the dependency between identity and expressions, at the expense of an increased complexity. However, in the 3D face recognition experiments we performed, the bilinear model had a worse identification rate than the linear model, and we decided to use the latter.

Acknowledgments

This work was funded by the Swiss National Science Foundation in the scope of the NCCR CO-ME project 5005-66380.

References

[1] B. Allen, B. Curless, and Z. Popović. The space of human body shapes: reconstruction and parameterization from range scans. In Rockwood [14].

[2] C. Basso and T. Vetter. Surface reconstruction via statistical models. In *Proceedings of 2nd International Conference on Reconstruction of Soft Facial Parts (RSFP 2005)*, Remagen, Germany, 17–18 March 2005.

[3] C. Basso, T. Vetter, and V. Blanz. Regularized 3d morphable models. In *Proceedings of the 1st IEEE International Workshop on Higher-Level Knowledge in 3D Modeling and Motion Analysis (HLK 2003)*, pages 3–11, Nice, France, 17 October 2003. IEEE Computer Society Press.

[4] V. Blanz, C. Basso, T. Poggio, and T. Vetter. Reanimating faces in images and video. *Computer Graphics Forum*, 22(3):641–641, 2003. Best Paper Award.

[5] V. Blanz, A. Mehl, T. Vetter, and H. P. Seidel. A statistical method for robust 3d surface reconstruction from sparse data. In *Int. Symp. on 3D Data Processing, Visualization and Transmission, Thessaloniki, Greece*, 2004.

[6] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26st International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH'99)*, pages 187–194. ACM Press, 1999.

[7] T. A. Davis. Umfpack version 4.3, 2004. http://www.cise.ufl.edu/research/sparse/umfpack/.

[8] I. Drori, D. Cohen-Or, and H. Yeshurum. Fragment-based image completion. In Rockwood [14], pages 303–312.

[9] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer-Verlag, 2001.

[10] K. Kähler, J. Haber, H. Yamauchi, and H.-P. Seidel. Head shop: Generating animated head models with anatomical structure. In S. Spencer, editor, *Proceedings of the 2002 ACM SIGGRAPH Symposium on Computer Animation*, pages 55–64, 2002.

[11] S. Marschner, B. Guenter, and S. Raghupathy. Modeling and rendering for realistic facial animation. In *Proceedings of the 11th Eurographics Workshop on Rendering*, pages 231–242, 2000.

[12] L. Pocock, editor. *Proceedings of the 28st International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH2001)*. ACM Press, 2001.

[13] E. Praun, W. Sweldens, and P. Schröder. Consistent mesh parameterization. In Pocock [12], pages 179–184.

[14] A. P. Rockwood, editor. *Proceedings of the 30st International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH2003)*. ACM Press, 2003.

[15] R. Szeliski and S. Lavallée. Matching 3-d anatomical surfaces with non-rigid deformations using octree-splines. In *IEEE Workshop on Biomedical Image Analysis*, pages 144–153. IEEE Computer Society, 1994.

[16] M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society, Series B*, 21(3):611–622, 1999.

[17] T. Vetter, M. Jones, and T. Poggio. A bootstrapping algorithm for learning linear models of object classes. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, pages 40–46. IEEE Computer Society Press, 1997.

[18] D. Vlasic, M. Brand, H. Pfister, and J. Popović. Face transfer with multilinear models. In J. L. Mohler, editor, *Proceedings of the 32nd International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH2005)*. ACM Press, July 31–August 4 2005.

[19] L. Zhang, N. Snavely, B. Curless, and S. Seitz. Spacetime faces: High resolution capture for modeling and animation. In D. Slothower, editor, *Proceedings of the 31st International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH2004)*, pages 548–558. ACM Press, Agust 8–12 2004.