Efficient Spatiotemporal Grouping Using the Nyström Method

Charless Fowlkes[1], Serge Belongie[2] and Jitendra Malik[1]
[1]University of California, Berkeley – Berkeley, CA 94720
[2]University of California, San Diego – La Jolla, CA 92093
{fowlkes,malik}@eecs.berkeley.edu, sjb@cs.ucsd.edu

Abstract

Spectral graph theoretic methods have recently shown great promise for the problem of image segmentation, but due to the computational demands, applications of such methods to spatiotemporal data have been slow to appear. For even a short video sequence, the set of all pairwise voxel similarities is a huge quantity of data: one second of a 256×384 *sequence captured at* 30*Hz entails on the order of* 10^{13} *pairwise similarities. The contribution of this paper is a method that substantially reduces the computational requirements of grouping algorithms based on spectral partitioning, making it feasible to apply them to very large spatiotemporal grouping problems. Our approach is based on a technique for the numerical solution of eigenfunction problems known as the Nyström method. This method allows extrapolation of the complete grouping solution using only a small number of "typical" samples. In doing so, we successfully exploit the fact that there are far fewer coherent groups in an image sequence than pixels.*

1 Introduction

The Gestalt school introduced several cues that are important to visual grouping including proximity, similarity, and common fate. Approaching the problem of grouping from a computational standpoint requires operationalizing such cues and combining them in an integrated framework. One method for combining both static image cues and motion information is to consider all images in a video sequence as a space-time volume and attempt to partition this volume into regions that are coherent with respect to the various grouping cues. This perspective is supported by evidence from psychophysics [9] that suggests spatial and temporal cues are treated jointly in the human visual system. The insight of considering a video signal as three dimensional for purposes of analysis goes back to Adelson and Bergen [1] and Baker et al. [4]. Volumetric segmentation has also been treated extensively in the literature on MRI processing [3], however, this domain lacks the causal structure (in the linear systems sense) possessed by video and doesn't consider cues that are unique to motion such as common fate.

Unified treatment of the spatial and temporal domains is also appealing as it could solve some of the well known problems in grouping schemes based on motion alone (e.g. layered motion models [23, 22]). For example, color or brightness cues can help to segment untextured regions for which the motion cues are ambiguous and contour cues can impose sharp boundaries where optical flow algorithms tend to drag along bits of background regions.

One computational framework for grouping within the space-time volume is to compute a k-way partitioning of a weighted graph where each node represents a volume unit (voxel) and the edge weights encode affinity between the voxels. Approaches in this framework have been developed and applied extensively to spatial segmentation of single images [19, 12, 8, 15, 14]. Unfortunately such successes have been slow to carry over to the case of spatiotemporal data. [1] Indeed, the conclusions of a recent panel discussion on spatiotemporal grouping [5] are that approaches in which the image sequence is treated as a multidimensional volume in x, y, t hold the greatest promise, but that efforts along these lines have been hampered largely by computational demands. The contribution of this paper is aimed directly at ameliorating this computational burden, thus making it feasible to extend the ideas of powerful pairwise grouping methods to the domain of video.

We formulate the grouping problem in the normalized cut (NCut) framework [19] which requires the solution of an $n \times n$ eigenproblem, where n is the total number of voxels in the space-time volume. (For example, $n \approx 3 \times 10^6$ for one second of a 256×384 image sequence captured at 30Hz.) Our approach to taming the computational demands of this problem is based on an approximation technique known as the Nyström method, originally developed for the numerical solution of eigenfunction problems. In short, this approach exploits the fact that the number of coherent groups in an

[1] Some preliminary steps in this direction were made by [18].

image sequence is considerably smaller than the number of voxels. It does so by extrapolating the complete grouping solution using the solution to a much smaller problem based on a few random samples drawn from the image sequence.

The structure of this paper is as follows. In Section 2 we discuss grouping cues and review the NCut grouping algorithm. We highlight our application of the Nyström method to the NCut grouping formulation in Section 3. Results are discussed in Section 4 and we conclude and ponder future work in Section 5.

2 Framework for Spatiotemporal Grouping

2.1 Spatiotemporal Grouping Cues

We would like to identify prominent groups within a space-time volume. In order to do so, it is first necessary to compute a measure of affinity between each unit of volume (voxel). Taking our cue from the Gestalt school, we consider proximity, similarity and common fate. We attach three features to each voxel in the sequence: location (x, y, t), intensity and color (L, a, b), and an optical flow vector (u, v) estimated between subsequent pairs of frames. We then compute the affinity between point i and j as

$$K_{ij} = e^{-\frac{1}{2}(\boldsymbol{x}_i-\boldsymbol{x}_j)^T\Sigma^{-1}(\boldsymbol{x}_i-\boldsymbol{x}_j)}$$

where $\boldsymbol{x}_i$ is the feature vector associated with the ith point in the image and Σ is a diagonal matrix whose entries are free parameters of the algorithm. The elements of K take on values between 0 and 1 which indicate how likely it is that two voxels belong to the same group. The diagonal entry of Σ associated with each cue is based on the expected variation within a group.

2.2 Partitioning with Normalized Cuts

Once the appropriate affinity function has been chosen, we would like to find a partitioning of the voxels into groups where each group has strong within group affinity and weak between group affinity. We employ the multiple eigenvector version of NCut [10] which embeds the voxels into a low dimensional Euclidean space such that significant differences in the normalized affinities are preserved while noise is suppressed. The $k-$means algorithm can then be used to discover groups of voxels that belong to the same region.

To find an embedding, we compute the matrix of eigenvectors V and eigenvalues Λ of the system

$$(D^{-1/2}KD^{-1/2})V = V\Lambda$$

where D is a diagonal matrix with entries $D_{ii} = \sum_j K_{ij}$. The the ith embedding coordinate of the jth voxel is then given by

$$E_{ij} = \frac{V_{i+1,j}}{\sqrt{(1-\lambda_{i+1})D_{jj}}}$$

where the eigenvectors have been sorted in ascending order by eigenvalue.

Unfortunately, the need to solve this system presents a serious computational problem. Since K grows as the square of the number of voxels in the sequence, for even very short video sequences it quickly becomes infeasible to fit K in memory, let alone compute its leading eigenvectors. One approach to this problem has been to use a sparse, approximate version of K in which each voxel is connected only to a few of its nearby neighbors in space and time and all other connections are assumed to be zero [18]. While this makes it possible to use efficient, sparse eigensolvers (i.e. Lanczos) the effects of this process are difficult to reason about. We propose an alternative approximation based on sampling in which we are able to keep all voxel similarities at the expense of some numerical accuracy in their values. Our approach also has the advantage of providing a clear quantification of the error introduced.

3 The Nyström Approximation

The Nyström method is a technique for finding numerical approximations to eigenfunction problems of the form:

$$\int_a^b K(x,y)\phi(y)dy = \lambda\phi(x)$$

We can approximate this integral equation by evaluating it at a set of evenly spaced points $\xi_1, \xi_2, \dots \xi_n$ on the interval $[a, b]$ and employing a simple quadrature rule,

$$\frac{(b-a)}{n}\sum_{j=1}^{n} K(x,\xi_j)\hat{\phi}(\xi_j) = \lambda\hat{\phi}(x) \qquad (1)$$

where $\hat{\phi}(x)$ is an approximation to the true $\phi(x)$. To solve (1) we set $x = \xi_i$ yielding the system of equations

$$\frac{(b-a)}{n}\sum_{j=1}^{n} K(\xi_i,\xi_j)\hat{\phi}(\xi_j) = \lambda\hat{\phi}(\xi_i) \quad \forall i \in \{1 \dots n\}$$

Without loss of generality, we let $[a, b]$ be $[0, 1]$ and structure the system as the matrix eigenvalue problem:

$$K\hat{\Phi} = n\hat{\Phi}\Lambda$$

where $K_{ij} = K(y_i, y_j)$ is the Gram matrix and $\Phi = [\boldsymbol{\phi}_1\boldsymbol{\phi}_2 \dots \boldsymbol{\phi}_n]$ are n approximate eigenvectors with corresponding eigenvalues $\lambda_1, \lambda_2, \dots \lambda_n$. Substituting back into equation (1) yields the *Nyström extension* for each $\hat{\phi}_i$

$$\hat{\phi}_i(x) = \frac{1}{n\lambda_i}\sum_{j=1}^{n} K(x,\xi_j)\hat{\phi}_i(\xi_j) \qquad (2)$$

3.1 Approximating the Eigenvectors of Affinity Matrices

The preceding analysis suggests that it should be possible to find approximate eigenvectors of a large Gram matrix by solving a much smaller eigenproblem using only a subset of the entries and employing the Nyström extension to fill in the rest. This is indeed the case. In this section we show an alternate analysis which relies purely on matrices and provides additional insight about the nature of the approximation.

Consider a Gram matrix $K \in \mathbb{R}^{p\times p}$ partitioned as follows

$$K = \begin{bmatrix} A & B \\ B^T & C \end{bmatrix} \tag{3}$$

with $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, and $C \in \mathbb{R}^{m\times m}$ where we will take n to be much smaller than m. Since K is positive definite, we can write it as the inner product of a matrix Z with itself: $K = Z^TZ$. If K is of rank n and the rows of the submatrix $[A \quad B]$ are linearly independent, Z can be written using only A and B as follows. Let Z be partitioned $Z = [X \quad Y]$ with $X \in \mathbb{R}^{p\times n}$ and $Y \in \mathbb{R}^{p\times m}$. Rewriting K we have:

$$K = Z^TZ = \begin{bmatrix} X^TX & X^TY \\ Y^TX & Y^TY \end{bmatrix}$$

Putting this in correspondence with (3) gives $A = X^TX$ and $B = X^TY$. Using the diagonalization $A = U\Lambda U^T$, where $U^TU = I$ we obtain

$$\begin{aligned} X &= \Lambda^{1/2}U^T \\ \hat{Y} &= (X^T)^{-1}B = \Lambda^{-1/2}U^TB \end{aligned}$$

Combining the two into $\hat{Z} = [X \quad \hat{Y}] \in \mathbb{R}^{p\times p}$ gives us

$$\begin{aligned} \hat{K} &= \begin{bmatrix} X^TX & X^T\hat{Y} \\ \hat{Y}^TX & \hat{Y}^T\hat{Y} \end{bmatrix} \\ &= \begin{bmatrix} X^TX & X^T\Lambda^{-1/2}U^TB \\ (\Lambda^{-1/2}U^T)^TX & (\Lambda^{-1/2}U^TB)^T\Lambda^{-1/2}U^TB \end{bmatrix} \\ &= \begin{bmatrix} A & B \\ B^T & B^TA^{-1}B \end{bmatrix} \end{aligned}$$

If the rank of K is greater than n or we fail to choose independent rows, then $\hat{K}$ is an approximation to K whose quality can be quantified as the norm of the Schur complement $\|C - B^TA^{-1}B\|$. The size of this norm is governed by the extent to which C is spanned by the rows of B.

Given this expression for $\hat{K}$, the approximate eigenvectors of K can be written in matrix form. Using again the diagonalization $A = U\Lambda U^T$, we have

$$\hat{K} = \bar{U}\Lambda\bar{U}^T, \quad \text{with} \quad \bar{U} = \begin{bmatrix} U \\ B^TU\Lambda^{-1} \end{bmatrix}$$

```
d1 = sum([A;B'],1);
d2 = sum(B,1) + sum(B',1)*inv(A)*B;
dhat = sqrt(1./[d1 d2])';
A = A.*(dhat(1:n)*dhat(1:n)');
B = B.*(dhat(1:n)*dhat(n+(1:m))');
Asi=sqrtm(inv(A));
Q=A+Asi*B*B'*Asi;
[U,L,T]=svd(Q);
V=[A;B']*Asi*U*inv(sqrt(L));
for i = 2:nvec+1
   E(:,i-1) = V(:,i)./V(:,1);
   E(:,i-1) = E(:,i-1)/sqrt(1-L(i,i));
end
```

Figure 1. Example MATLAB code for finding the first `nvec` embedding vectors of the normalized affinity matrix given unnormalized submatrices `A` of size `n×n` and `B` of size `n×m`.

The lower block of $\bar{U}$ is clearly just matrix notation for the repeated application of the Nyström extension as given in equation (2). The only remaining detail is that the columns of $\bar{U}$ are not necessarily orthogonal. This is addressed as follows. Let $A^{1/2}$ denote the symmetric positive definite square root of A, define $Q = A + A^{-1/2}BB^TA^{-1/2}$ and diagonalize it as $Q = R\hat{\Lambda}R^T$. Now define the matrix $\hat{V}$ as

$$\hat{V} = \begin{bmatrix} A \\ B^T \end{bmatrix} A^{-1/2}R\hat{\Lambda}^{-1/2} \tag{4}$$

It can then be shown that $\hat{V}$ and $\hat{\Lambda}$ diagonalize $\hat{K}$, i.e. $\hat{K} = \hat{V}\hat{\Lambda}\hat{V}^T$ and $\hat{V}^T\hat{V} = I$. Due to lack of space we omit the proof.

3.2 Approximate Normalized Cuts

To apply the matrix form of the Nyström extension to NCuts, it is necessary to compute the row sums of $\hat{K}$. This is possible without explicitly evaluating the $B^TA^{-1}B$ block since

$$\begin{aligned} \hat{\boldsymbol{d}} = \hat{K}\mathbf{1} &= \begin{bmatrix} A\mathbf{1}_m + B\mathbf{1}_n \\ B^T\mathbf{1}_m + B^TA^{-1}B\mathbf{1}_n \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{a}_r + \boldsymbol{b}_r \\ \boldsymbol{b}_c + B^TA^{-1}\boldsymbol{b}_r \end{bmatrix} \end{aligned} \tag{5}$$

where $\boldsymbol{a}_r, \boldsymbol{b}_r \in \mathbb{R}^m$ denote the row sums of A and B, respectively, and $\boldsymbol{b}_c \in \mathbb{R}^n$ denotes the column sum of B.

With $\hat{\boldsymbol{d}}$ in hand, the blocks of $\hat{D}^{-1/2}\hat{K}\hat{D}^{-1/2}$ that are needed to approximate the leading eigenvectors are given as

$$A_{ij} \leftarrow \frac{A_{ij}}{\sqrt{\hat{\boldsymbol{d}}_i\hat{\boldsymbol{d}}_j}}, \qquad i,j = 1,\ldots,m$$

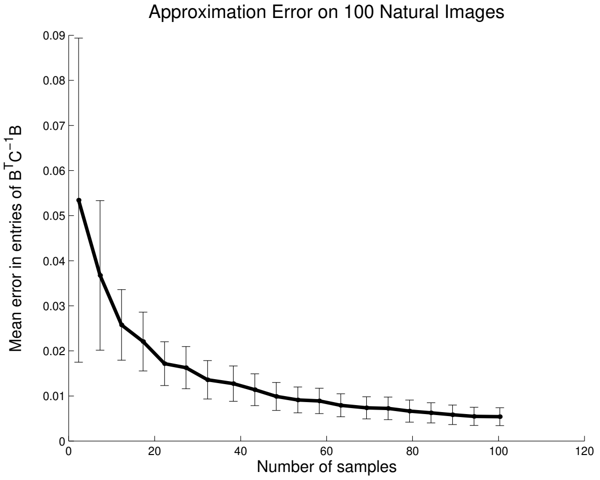

Figure 2. The behavior of the approximation error for natural images with increasing numbers of samples. The cues used here are color, intensity and proximity. The error bars indicate one standard deviation taken over a set of 100 natural images chosen from the Corel database. Recall that the entries of K range from 0 to 1 so error values shown here are on the order of a few percent.

and

$$B_{ij} \leftarrow \frac{B_{ij}}{\sqrt{\hat{\boldsymbol{d}}_i \hat{\boldsymbol{d}}_{j+m}}}, \qquad i = 1, \ldots, m, j = 1, \ldots, n$$

All that remains is to apply equation (4) as before. The entire procedure for computing the normalized embedding coordinates given A and B is outlined in Figure 1 in the form of some simple MATLAB code.

3.3 Computational Demands

Once the affinities $[A \quad B]$ have been computed, the most expensive operation is the diagonalization of A and Q. These operations scale as $O(n^3)$ where n is the number of samples employed in the approximation.

How many samples are required to achieve a good partitioning? The answer in practice seems to be very few. We studied the approximation quality of an affinity matrix consisting of color and proximity cues for a set of one hundred 480×320 natural images chosen from the Corel database of stock photos. We chose proximity and color parameters $\sigma_{prox} = 400$ and $\sigma_{color} = 0.01$ and the sample coordinates were chosen uniformly at random. Since it's not feasible to hold C in memory, the error was estimated by considering submatrices of C. Figure 2 demonstrates the fall-off in error as the number of samples are increased. The error bars indicate one standard deviation over the set of images. Without providing a perturbation-theoretic argument, we note that the subjective quality of the eigenvectors follows a similar pattern as one might expect.

A simple analysis of this rapid decay goes as follows. In the limiting case that the affinity function is a perfect indicator of whether two points lie in the same segment, then a single sample from each segment would be sufficient to span the rows of K. This clearly provides leverage to the intuition that segmentation should scale with the number of segments rather than the number of pixels in the image.

3.4 Related Work on Approximation

E. J. Nyström published his method in the late 1920's [11]. Its use in approximating solutions to integral equations is well known for its simplicity and accuracy [2, 6, 13]. The Nyström method has also been recently applied in the kernel learning community [24] for fast approximate Gaussian process classification and regression. As noted in [24], this approximation method directly corresponds to the kernel PCA feature space projection technique of [17]. The authors of [20] present a greedy method for selecting the best rows/columns to use for the approximation. A related work in the area of document analysis is that of [7], wherein only the off-diagonal blocks of the affinity matrix are known, i.e. only bipartite weights are available. The author then applies the Normalized Cut method, which reduces to a simple SVD on the non-zero blocks, in order to accomplish "co-clustering" of documents and keywords.

4 Results

We provide several examples of video segmentation using our algorithm. Each of the results shown make use of 100 samples drawn at random from the first, middle and last frame in the sequence. Figure 3 shows the performance of our algorithm on the flower garden sequence. A proper treatment would require dealing with the texture in the flowerbed and the illusory contours that define the tree trunk. However, the discontinuities in local color and motion alone are enough to yield a fairly satisfying segmentation.

Figure 4 demonstrates segmentation of a relatively unclutered scene. Processing the entire sequence as a volume automatically provides correspondences between segments in each frame. We note that using motion alone would tend to track the shadows and specularities present on the background and fail to find the sharp boundaries around the body. Figure 5 shows performance in a more complicated sequence involving multiple moving objects in addition to camera translation. On a 800MHz Pentium III processor, segmenting a $120 \times 120 \times 5$ voxel sequence (i.e. Figure 5) takes less than 1 minute in MATLAB.

5 Conclusion

We have introduced an approximate version of NCut based on the Nyström method which makes it possible to solve very large grouping problems efficiently. We have demonstrated the application of this technique to spatiotemporal data with encouraging results. By simultaneously making use of both static cues (color, intensity, location in the image) and dynamic cues (optical flow, location in time) we are able to find coherent groups within a variety of video sequences.

More work is clearly needed in order to achieve high quality segmentation on general video. Of key importance is the incorporation of more sophisticated grouping cues and gating mechanisms. For example, there are many static image cues that can be extended to the domain of video. If the boundary of a region is indicated by a strong contour, it will sweep out a surface in the space-time volume. Voxels that are on opposite sides of such an intervening surface shouldn't be as likely to belong to the same group. Likewise, texture has a space-time equivalent in the form of dynamic textures (e.g. tree leaves blowing in the wind) [21, 16].

Our hope is that the approximation method we have presented will facilitate the further development of segmentation methods which work directly on the spatio-temporal volume.

6 Acknowledgements

The authors would like to acknowledge Alice Zheng, Andrew Ng, William Kahan, and Gianluca Donato for helpful discussions and valuable input during the course of this work. This work was supported in part by a UC Berkeley MICRO Fellowship (to CF).

References

[1] E. H. Adelson and J. R. Bergen. Spatiotemporal energy models for the perception of motion. *J. Opt. Soc. Am. A*, 2(2):284–299, 1985.

[2] C. T. H. Baker. *The numerical treatment of integral equations*. Oxford: Clarendon Press, 1977.

[3] J. Bezdek, L. Hall, and L. Clarke. Review of MR image segmentation techniques using pattern recognition. *Medical Physics*, 20(4):1033–1048, 1993.

[4] R. Bolles, H. Baker, and D. Marimont. Epipolar-plane image analysis: an approach to determining structure from motion. *Int. Journal of Computer Vision*, 1:7–55, 1987.

[5] K. Boyer, D. Fagerström, M. Kubovy, P. Johansen, and S. Sarkar. POCV99 breakout session report: Spatiotemporal grouping. In S. Sarkar and K. L. Boyer, editors, *Perceptual Organization for Artificial Vision Systems*. Kluwer Academic Publishers, 2000.

[6] L. Delves and J. Mohamed. *Computational Methods for Integral Equations*. Cambridge University Press, 1985.

[7] I. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. Technical Report 2001-05, UT Austin CS Dept., 2001.

[8] Y. Gdalyahu, D. Weinshall, and M. Werman. Stochastic image segmentation by typical cuts. In *CVPR*, pages 596–601, 1999.

[9] S. Gepshtein and M. Kubovy. The emergence of visual objects in space-time. *Proc. Nat. Acad. Sci. USA*, 97(14):8186–8191, 2000.

[10] J. Malik, S. Belongie, T. Leung, and J. Shi. Contour and texture analysis for image segmentation. *Int. Journal of Computer Vision*, 43(1):7–27, June 2001.

[11] E. J. Nyström. Über die praktische auflösung von linearen integralgleichungen mit anwendungen auf randwertaufgaben der potentialtheorie. *Commentationes Physico-Mathematica*, 4(15):1–52, 1928.

[12] P. Perona and W. T. Freeman. A factorization approach to grouping. In *ECCV*, pages 655–670, 1998.

[13] W. Press, S. Teukolsky, W. Vetterling, and B.P.Flannery. *Numerical Recipies in C, 2nd Edition*. Cambridge University Press, 1992.

[14] J. Puzicha, T. Hofmann, and J. Buhmann. Non-parametric similarity measures for unsupervised texture segmentation and image retrieval. In *CVPR*, pages 267–272, 1997.

[15] S. Sarkar and K. Boyer. Quantitative measures of change based on feature organization: Eigenvalues and eigenvectors. In *CVPR*, pages 110–136, 1996.

[16] A. Schödl, R. Szeliski, D. H. Salesin, and I. Essa. Video textures. In *SIGGRAPH*, pages 489–498, 2000.

[17] B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998.

[18] J. Shi and J. Malik. Motion segmentation and tracking using normalized cuts. In *ICCV*, pages 1154–1160, 1998.

[19] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. PAMI*, 22(8):888–905, August 2000.

[20] A. Smola and B. Schölkopf. Sparse greedy matrix approximation for machine learning. In *ICML*, pages 911–918, 2000.

[21] S. Soatto, G. Doretto, and Y. Wu. Dynamic textures. In *ICCV*, pages 439–446, 2001.

[22] Y. Weiss. Smoothness in layers: Motion segmentation using nonparametric mixture estimation. In *CVPR*, pages 520–526, 1997.

[23] Y. Weiss and E. Adelson. A unified mixture framework for motion segmentation: incorporating spatial coherence and estimating the number of models. In *CVPR*, pages 321–326, 1996.

[24] C. Williams and M. Seeger. Using the Nyström method to speed up kernel machines. In *NIPS-13*, pages 682–688, 2001.

Figure 3. The Flower Garden Sequence: Each column represents our segmentation of a frame from the sequence of four images shown in the top row. Each row shows slices through a space-time segment. It's important to note that the algorithm provides segment correspondence between frames automatically. The image dimensions are 120×80 pixels.

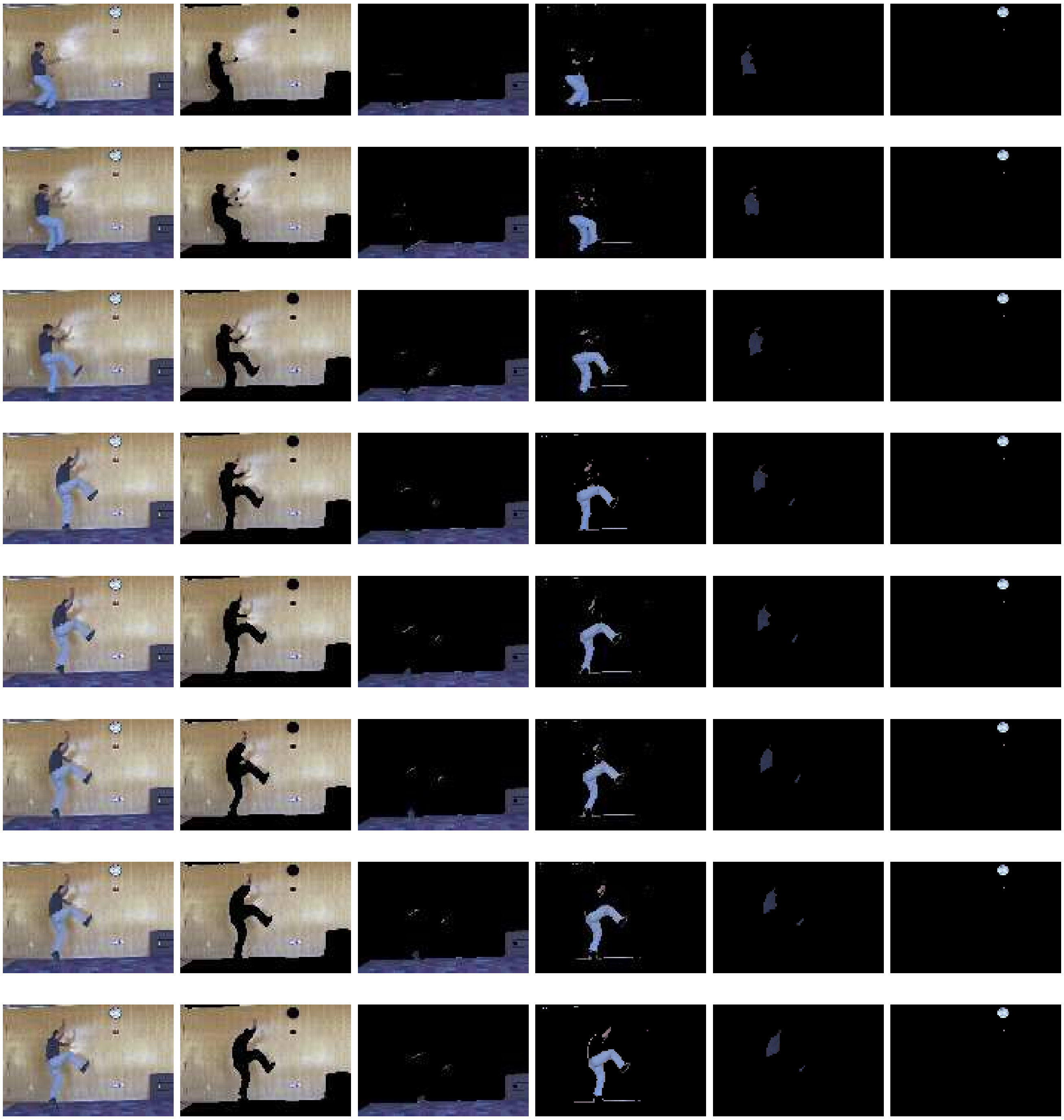

Figure 4. The Leap: The original frames (120 × 80 pixels) are shown in the left column. Each column shows slices through a space-time segment.

Figure 5. The Firetruck: The original 120×120 pixel sequence is shown across the top row. The remaining rows indicate individual segments.