Sequential Factorization: An Algorithmic Approach to Structure from Motion

Natalie Bishay
Stanford University
Department of Electrical Engineering
nbishay@stanford.edu

Abstract

The factorization method, as presented by Tomasi and Kanade, produces a robust and accurate reconstruction of the three-dimensional scene when at least two or more frames, or images, are taken. The well-known method utilizes the singular value decomposition, or SVD, to compute the structure and motion matrices. However, in real-time applications, the repeated computation of the SVD after acquiring all frames may not be tractable computationally. For this reason, the sequential factorization method is proposed. The sequential factorization method seeks to leverage the fact that only the three dominant eigenvectors are necessary in computing the structure and motion matrices. Doing this update can speed up the process to occur in O(P^2) time, as opposed to O(FP^2) time. The experimental results demonstrate that we get reconstruction that is almost exactly like the original method. This method was originally implemented by Morita and Kanade, and the methods discussed in this report are almost entirely from their work.

1. Introduction

The structure from motion problem is a very important one in many applications, such as robotics, medical device imaging, and much more. The primary goal of the structure from motion problem is to acquire the three-dimensional layout of a scene from multiple frames, or images, of a scene from different perspective. With these frames, we can correctly estimate the depth, height, and width of all the feature points in all of the images, and then re-project them to recreate a 3D model of the original scene. The factorization method, which was proposed by Tomasi and Kanade, utilizes the singular value decomposition, or SVD, to compute these structure and motion matrices, where the points reconstructed in the scene come from the structure matrix.

The factorization method is an incredibly robust method, with very high scene accuracy (if there are minimal occlusions). However, the factorization has limitations. One of these limitations is the time it takes to perform the singular value decomposition. The time the SVD takes is $O(mn\min(m,n))$ for an m by n matrix[3], which, for F frames and P feature points ends up being $O(FP^2)$. But in the sequential factorization method, only $O(P^2)$ operations are necessary. Another limitation of the factorization method is that it requires all of the feature points and frames before computing, which can slow down many real-time processes. However, the sequential factorization method views the sequence of points per frame as a vector time series, which allows for the algorithms discussed to just "add on" to the work done at the previous frame instead of waiting until all the frames are obtained to begin. This can be extremely advantageous in real-time processes, such as robotics.

2. Background/Related Work

There are two prior works necessary to the completion of this report, and the first is the already-mentioned factorization method, which is necessary to discuss as it is our point of comparison. The second is the prior work on sequential factorization.

2.1 The Original Factorization Method

This work was done by Tomasi and Kanade[1]. It starts with a matrix $W$of size $2F \times P$. The reason it is $2F$ is because each of the P feature points have an x and a y coordinate. The W matrix is constructed as follows:

$$W = \begin{bmatrix} x_1^T \\ \vdots \\ x_f^T \\ y_1^T \\ \vdots \\ y_f^T \end{bmatrix} (1)$$

Tomasi and Kanade state that $W$is highly rank-deficient, because all of the points in W are just homographies of each other with respect to the centroid. Precisely, W is rank 3 if noiseless. If $W = MS$, then, the matrix M and S must also, at most, be rank 3. M is a $2F \times 3$ motion matrix, and S is a $3 \times P$ structure matrix.

To actually compute M and S from W, Tomasi and Kanade proposed the singular value decomposition:

$$W = U\Sigma V^T \ (2)$$

One of the strengths of the singular value decomposition as compared to other decomposition methods is the ability to create lower-rank approximations of our matrix by setting desired values of the diagonal matrix Σ to 0. Since, in practice, performing the singular value decomposition on noisy values in W may lead to higher-rank matrices, we can set values in the diagonal of Σ besides the first 3 singular values equal to zero.

We then can see that a few appropriate equalities arise, and the one suggested by Tomasi and Kanade is as follows:

$$\hat{M} = \mathrm{U}\Sigma^{1/2} \ (3)$$

and

$$\hat{S} = \Sigma^{1/2}V^T \ (4)$$

Now, according to the paper by Morita and Kanade, as well on the aforementioned paper on the singular value decomposition, we know that the left singular vectors (in U) span the column space, and the right singular vectors (in V) span the row space. We call these the motion space and the shape space, respectively. The shape space is of special importance, especially in the development and analysis of the algorithms that will soon be described, that depend heavily on the incremental update of the shape space.

2.2 The Sequential Factorization Method

Because most of this project was the understanding and implementation of the sequential factorization method, I will go through the high-level ideas, as well as orthogonal iteration, which is an essential part of the algorithm, and save the mechanics of the approach for the Approach section of this paper.

The high-level idea of the sequential factorization method is to iteratively compute the structure S_f at each frame, and the key insight to being able to do this is the fact that the structure space is stationary. Stationarity means that it doesn't change with time. We therefore, according to Morita and Kanade[2], should be able to compute S_f from S_{f-1} inexpensively. We will discuss how this happens a bit later.

2.2.1 Orthogonal Iteration

For this implementation, it is necessary to discuss how to compute p dominate eigenvectors of a symmetric matrix. Starting with a symmetric matrix C that is $n \times n$ and any $n \times p$ matrix Q_0 that has orthonormal columns, we perform the following iterations to converge to the eigenvalues of C:

for $k = 1, \ldots, p$:
  $Y_k = BQ_{k-1}$
  $Q_k R_k = Y_k$ (QR factorization)
end

The matrix Q_p will necessarily converge to the eigenvalues of C. According to Morita and Kanade, this method is especially advantageous in the case where C is large and only a small number of eigenvalues are needed.

3. Approach

Now we will enter into the details of the sequential factorization method, up to the metric transformation. We recall, from equation (1), that the matrix W is as follows:

$$W = \begin{bmatrix} x_1^T \\ \vdots \\ x_f^T \\ y_1^T \\ \vdots \\ y_f^T \end{bmatrix}$$

Let us now define a matrix time series Z, which is a symmetric matrix:

$$Z_f = W_f^T W_f = Z_{f-1} + x_f x_f^T + y_f y_f^T \quad (5)$$

Since $W = U\Sigma V$,

$Z = V\Sigma^2 V^T$, which we can see corresponds to the eigenvalue decomposition of a matrix

$$Z = Q\Lambda Q^T,$$

where we equate $V = Q$ and $\Lambda = \Sigma^2$. This means that the eigenvectors of Z_f correspond to V, which we recall is the basis for the shape space of W. Therefore, we can construct an algorithm that levereges this information similar to the one used to compute orthogonal iteration. Q_0 is $P \times 3$ matrix that has orthonormal columns, and Z_0 is a matrix initialized to all zeros of size $P \times P$. f is the iterator over all of the frames F:

for $f = 1, \ldots, F$:
  $Z_f = Z_{f-1} + x_f x_f^T + y_f y_f^T$
  $Y = Z_f Q_{f-1}$
  $Q_f R = Y$ (QR factorization)
end

The matrix Q_f converges to the eigenvalues of Z as mentioned before, which provides the shape space of W.

Consequently, we can see that from iteration to iteration, we are incorporating new feature points in every step. We must also note that if the range(V_f) changes, no convergence is expected, but since we know the range does not change since the structure is stationary, as mentioned before, we know that range(V_f) is stationary and range(Q_f) converges to it.

Morita and Kanade go on to reconstruct up to and including a metric transformation, but I didn't have to implement this. However, I found that the matrix Q computed accurately reconstructs the structure of the given points.

4. Experiments

We can both qualitatively and quantitatively see the results of this method by comparing the original factorization method and the sequential factorization method. First, it is necessary to introduce the distance metric utilized by Morita and Kanade to demonstrate the difference between range spaces. The distance metric used to compare how close two range spaces is defined by:

$$dist\left(range(Q_k),\, range(X_p)\right) = \left\|Q_kQ_k^T - X_pX_p^T\right\| \tag{6}$$

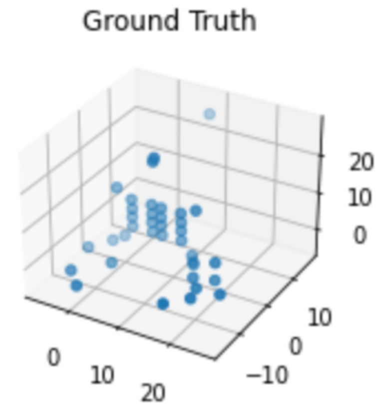
The estimation error of the sequential factorization method with respect to the true shape space is given by

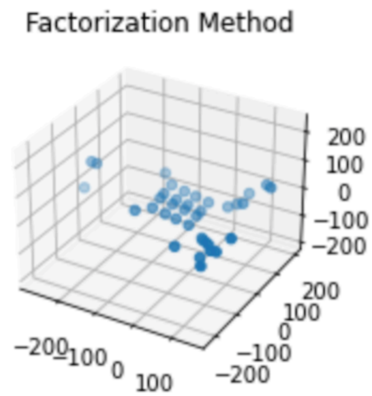
$$E_q = dist\left(range(Q_f),\, range(S^T)\right) \qquad (7).$$

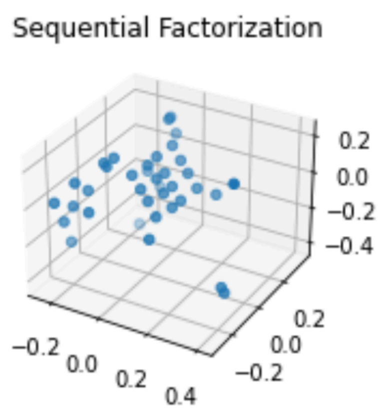
And the estimation error of the original factorization method is given by

$$E_0 = dist\left(range(V_f),\, range(S^T)\right) \qquad (8).$$

For my experiment, I used a set of 3D points randomly instantiated, yet with a clear form and shape that I wanted to reproduce, and re-projected them onto multiple different image frames, where the frames differed by an affine transformation. I was able to calculate a Q and plot it, and compare it to the original (ground truth), as well as to the original factorization method. The results are as follows:







Furthermore, I calculated that $E_q - E_0 = 0$. In other words, the errors given by the sequential method are almost identical to the errors generated by the original method. We can see that the sequential factorization method very accurately and robustly can reconstruct the structure of a stationary scene.

4. Conclusion

In conclusion, we can see that the sequential factorization method provides estimates of the shape at each frame for a sequence of images all capturing the same scene. We have seen that it accurately mimics the robust nature of the original factorization method. I learned that really simple linear algebra techniques can be leveraged to treat the feature points at each frame as a vector time series. The fact that symmetric matrices can be decomposed into eigenvalues is really the key fact here, and led to a simple yet elegant algorithm! However, I have not analyzed two significant aspects of this work that the original paper did, and could be done for further work.

Firstly, what's missing from these experiments is a time computational analysis, which time did not permit me to complete. One expects that the sequential factorization method would be much faster than the original factorization method, especially in the case where there are many frames and therefore many iterations; this is when the sequential factorization method should shine.

Secondly, as alluded to, a case where there are many frames is quite necessary to demonstrate where the sequential factorization would be best suited. This is due to the already-discussed time complexity of the sequential factorization method compared to the original factorization method, which is dominated by the singular value decomposition.

Thirdly, the paper by Morita and Kanade goes into details about the metric transformation needed to exactly replicated the given scene, and not just up to an affine ambiguity. This could also be done to demonstrate how close indeed the sequential factorization method and the original factorization method are.

But preliminarily, it is quite astounding the simple linear algebra that was leveraged to generate the image produced above. I hope to implement these improvements in the future, but for now, am quite impressed with the work done on this topic using simple linear algebra facts.

5. References

[1] Kanade, Takeo, and Daniel D. Morris. "Factorization Methods for Structure from Motion."

[2] Morita, T., and T. Kanade. "A Sequential Factorization Method for Recovering Shape and Motion from Image Streams." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 8, 1997.

[3] Vasudevan, Vinita, and M. Ramakrishna. "A Hierarchical Singular Value Decomposition Algorithm for Low Rank Matrices." *ArXiv*, 10 May 2019.

6. Supplementary Materials

https://github.com/natbishay/CS231A-Project