ITERATIVE RIGID BODY TRANSFORMATION ESTIMATION FOR VISUAL 3-D OBJECT TRACKING

Micha Hersch, Thomas Reichert and Aude Billard
LASA Laboratory, EPFL, 1015 Lausanne, Switzerland

Keywords: Stereo vision tracking, Rigid body transformation estimation.

Abstract: We present a novel yet simple 3D stereo vision tracking algorithm which computes the position and orientation of an object from the location of markers attached to the object. The novelty of this algorithm is that it does not assume that the markers are tracked synchronously. This provides a higher robustness to the noise in the data, missing points and outliers. The principle of the algorithm is to perform a simple gradient descent on the rigid body transformation describing the object position and orientation. This is proved to converge to the correct solution and is illustrated in a simple experimental setup involving two USB cameras.

1 INTRODUCTION

Estimating the 3-D rigid body transformation aligning two noisy sets of identifiable points is considered a solved problem in computer vision. Indeed, various closed form solutions have been suggested in the last two decades (Arun et al., 1987; Horn, 1987; Walker et al., 1991), and those solutions have been widely used and compared (Eggert et al., 1997). However, in spite of those existing solutions, we address once again this problem and suggest an iterative solution to the rigid body estimation problem. Our belief is that in many applications, an iterative solution is preferable to a closed-form solution, especially if the rigid body transformation changes in time, for example when tracking a moving object. The major reasons for this is that an iterative solution would be more robust to noise in the data and that and would not assume synchronicity of the set of points.

2 SETTING AND NOTATIONS

We consider a rigid body transformation $\mathbf{T}$ transforming a set of n vectors $\{\mathbf{x}_i\}$ into another set of n vectors $\{\mathbf{y}_i\}$. This transformation is described by a rotation $\mathbf{R}$ around an axis passing through the origin and a translation $\mathbf{V}$ by a vector $\mathbf{v}$:

$$\mathbf{y}_i = \mathbf{T}(\mathbf{x}_i) = \mathbf{R}(\mathbf{x}_i) + \mathbf{v}. \tag{1}$$

When considering a 3-D tracking application, the rigid body transformation $\mathbf{T}$ can be used to describe the position and orientation of the tracked object, relatively to a reference position and orientation. The reference positions of the n markers on the objects make the set of $\{\mathbf{x}_i\}$. The positions of those markers when tracked by a stereo vision system constitute the set of $\{\mathbf{y}_i\}$. It is assumed that the markers can be distinguished one from another, for example by using different colors. If the object is moving, the evolution of $\mathbf{T}$ yields the trajectory of the object.

3 ROTATIONS

In this paper, we use the spinor representation of rotations which is briefly recalled here, adopting the approach described in (Hestenes, 1999). This representation is very similar to the quaternion representation. The spinor $\overline{q}$ representing the rotation $\mathbf{R}$ is given by a scalar α and imaginary vector $\mathbf{b}i$. The direction of $\mathbf{b}$ yields the rotation axis (passing through the origin) and its norm is equal to $\sin(\theta/2)$, where θ is the rotation angle. The scalar α is given by $\cos(\theta/2)$.The rotation of a vector $\mathbf{x}$ by a spinor $\overline{q}$ is given by the following equation.

$$\mathbf{R}_\mathbf{b}(\mathbf{x}) = (1-2\mathbf{b}^T\mathbf{b})\mathbf{x} + 2\sqrt{(1-\mathbf{b}^T\mathbf{b})}\mathbf{b}\times\mathbf{x} + 2(\mathbf{b}^T\mathbf{x})\mathbf{b}, \tag{2}$$

where $\mathbf{R}_\mathbf{b}$ denotes the rotation represented by $\mathbf{b}$.

Hersch M., Reichert T. and Aude A. (2008).
ITERATIVE RIGID BODY TRANSFORMATION ESTIMATION FOR VISUAL 3-D OBJECT TRACKING.
In *Proceedings of the Third International Conference on Computer Vision Theory and Applications*, pages 674-677
DOI: 10.5220/0001087106740677
Copyright © SciTePress

4 ITERATIVE ESTIMATION OF A RIGID BODY TRANSFORMATION

We now present to the algorithm for iteratively estimating a rigid body transformation given a set of n points $\{\mathbf{x}_i\}$ and its noisy transform $\{\mathbf{y}_i\}$. The principle of the algorithm is quite trivial. Starting from an initial guess for the parameters **b** and **v** of the transformation, it consists simply on a gradient descent on the squared distance between the measurement $\mathbf{y}_i$ and the transformed point $\mathbf{T}_{\mathbf{b},\mathbf{v}}(\mathbf{x}_i)$

$$\Delta\mathbf{b} = -\varepsilon\frac{\partial}{\partial\mathbf{b}}\frac{1}{2}\left|\left|\mathbf{y}_i - \mathbf{T}_{\mathbf{b},\mathbf{v}}(\mathbf{x}_i)\right|\right|^2 \tag{3}$$

$$\Delta\mathbf{v} = -\varepsilon\frac{\partial}{\partial\mathbf{v}}\frac{1}{2}\left|\left|\mathbf{y}_i - \mathbf{T}_{\mathbf{b},\mathbf{v}}(\mathbf{x}_i)\right|\right|^2, \tag{4}$$

where ε is the learning rate. One assumes that i takes values from 1 to n in a uniformly distributed manner. So at each time step, an index i is selected among the available points and **b** and **v** are update according to (3) and (4).
The actual development of those two equations yields:

$$\begin{aligned}\Delta\mathbf{b} &= 2\varepsilon\big(\mathbf{y}_i - \mathbf{T}_{\mathbf{b},\mathbf{v}}(\mathbf{x}_i)\big)^T\Big(-2\mathbf{x}_i\mathbf{b}^T - \\ &\quad \frac{1}{\sqrt{(1-\mathbf{b}^T\mathbf{b})}}(\mathbf{b}\times\mathbf{x}_i)\mathbf{b}^T + 1\sqrt{(1-\mathbf{b}^T\mathbf{b})}\mathbf{x}_i\uparrow + \\ &\quad \mathbf{b}\mathbf{x}_i^T + (\mathbf{b}^T\mathbf{x}_i)\mathcal{I}\Big) && (5)\\ \Delta\mathbf{v} &= \varepsilon\big(\mathbf{y}_i - \mathbf{T}_{\mathbf{b},\mathbf{v}}(\mathbf{x}_i)\big), && (6)\end{aligned}$$

where $\mathcal{I}$ is the 3×3 identity matrix and the unary operator $\uparrow$ is defined as

$$\mathbf{x}\!\uparrow\doteq\frac{\partial}{\partial\mathbf{b}}(\mathbf{b}\times\mathbf{x}) = \begin{pmatrix} 0 & x^{(3)} & -x^{(2)} \\ -x^{(3)} & 0 & x^{(1)} \\ x^{(2)} & -x^{(1)} & 0 \end{pmatrix}, \tag{7}$$

with $\mathbf{x} = [x^{(1)}\; x^{(2)}\; x^{(3)}]^T$.

This concludes the description of the algorithm. For efficiency purposes, it is preferable to choose reference positions so that the $\mathbf{x}_i$ are centered on the origin. This allows to reduce the influence of **b** on the computation of **v**.

5 CONVERGENCE

In this section, we prove that if there exists a rigid body transformation matching the two sets of points $\{\mathbf{x}_i\}$ to $\{\mathbf{y}_i\}$, then the iterative algorithm described above will converge to it.

Let $\mathbf{T}^*$ be the true transformation mapping a finite set of points $\{\mathbf{x}_i\} = \mathcal{V}$ into their corresponding image. If $\mathcal{V}$ contains at least three unaligned points, there is only one such transformation. Let $\mathbf{T} \neq \mathbf{T}^*$ be the current estimate of this transformation.
We then define the following function $\mathbf{E}(\mathbf{T})$

$$\mathbf{E}(\mathbf{T}) = \sum_{i=1}^{n}\mathbf{E}_i(\mathbf{T}), \text{ with } \mathbf{E}_i(\mathbf{T}) = \frac{1}{2}\|\mathbf{T}\mathbf{x}_i - \mathbf{T}^*\mathbf{x}_i\|^2 \tag{8}$$

Here and in the rest of this paper, the parentheses around $\mathbf{x}_i$ are omitted to lighten the notation. We also define the vector $\mathbf{p} = [\mathbf{b}^T\mathbf{v}^T]^T$ to be the vector parameterizing the transformation.

We first show that the algorithm always converges to a solution. If ε tends to zero and t is the time, then $\varepsilon^{-1}\Delta\mathbf{b}$ and $\varepsilon^{-1}\Delta\mathbf{v}$ tend respectively to $\frac{\partial}{\partial t}\mathbf{b}$ and $\frac{\partial}{\partial t}\mathbf{v}$. So the gradient descent of the algorithm means that $\frac{\partial}{\partial t}\mathbf{p} = -\frac{\partial}{\partial\mathbf{p}}\mathbf{E}_i(\mathbf{T})$. We thus have

$$\begin{aligned}\frac{\partial}{\partial t}\mathbf{E} = \frac{\partial}{\partial t}\sum_{i=1}^{n}\mathbf{E}_i &= \sum_{i=1}^{n}\frac{\partial}{\partial t}\mathbf{E}_i = \sum_{i=1}^{n}\frac{\partial}{\partial\mathbf{p}}\mathbf{E}_i\frac{\partial}{\partial t}\mathbf{p} = \\ \sum_{i=1}^{n}\frac{\partial}{\partial\mathbf{p}}\mathbf{E}_i\Big(-\frac{\partial}{\partial\mathbf{p}}\mathbf{E}_i\Big) &= \sum_{i=1}^{n}-\Big(\frac{\partial}{\partial\mathbf{p}}\mathbf{E}_i\Big)^2 \leq 0\end{aligned}$$

The function $\mathbf{E}(\mathbf{T})$, being positive, the algorithm always converges to a solution. It remains to be shown that this solution is correct.

In order to show that the algorithm converges to the right solution $\mathbf{T}^*$, we show that for any $\mathbf{T}$, $\mathbf{T}^*$, $\mathcal{V}$, satisfying the conditions mentioned above, there is a transformation $\mathbf{T}^\dagger$ belonging to a neighborhood of $\mathbf{T}$ such that

$$\mathbf{E}(\mathbf{T}^\dagger) < \mathbf{E}(\mathbf{T}) \tag{9}$$

This amounts to saying that there is no local minimum for $\mathbf{E}(\mathbf{T})$. We assume, without loss of generality, that the $\mathbf{x}_i$ are centered. Let us consider the transformation $\mathbf{T}^\dagger$ defined by translation vector $\mathbf{v}^\dagger$ and rotation $\mathbf{R}^\dagger$

$$\begin{aligned}\mathbf{v}^\dagger &= \mathbf{v} + \varepsilon(\mathbf{v}^* - \mathbf{v}) && (10)\\ \mathbf{R}^\dagger &= \varepsilon\mathbf{R}^+ \circ \mathbf{R} \quad \text{with} \quad \varepsilon > 0. && (11)\end{aligned}$$

In the above expression $\varepsilon\mathbf{R}^+$ is an infinitesimal rotation of unit rotation axis given by

$$\mathbf{b}^+ = z\sum_i \mathbf{R}\mathbf{x}_i \times \mathbf{R}^*\mathbf{x}_i \tag{12}$$

where $z = \|\sum_i \mathbf{R}\mathbf{x}_i \times \mathbf{R}^*\mathbf{x}_i\|^{-1}$. This means that $\mathbf{R}^\dagger$ is in the neighborhood of $\mathbf{R}$. If ε is small enough, we have, see (Altmann, 1986),

$$\mathbf{R}^\dagger\mathbf{x} = \mathbf{R}\mathbf{x} + \varepsilon(\mathbf{b}^+ \times \mathbf{R}\mathbf{x}). \tag{13}$$

Thus the variation in $\mathbf{E}$ when moving from $\mathbf{T}$ to $\mathbf{T}^\dagger$ is given by

$$
\begin{aligned}
\Delta\mathbf{E} = {} & \mathbf{E}(\mathbf{T}^\dagger) - \mathbf{E}(\mathbf{T}) \qquad (14)\\
= {} & \sum_i \|\mathbf{T}^\dagger\mathbf{x}_i - \mathbf{T}^*\mathbf{x}_i\|^2 - \sum_i \|\mathbf{T}\mathbf{x}_i - \mathbf{T}^*\mathbf{x}_i\|^2\\
= {} & \sum_i \|\mathbf{T}^\dagger\mathbf{x}_i\|^2 - \|\mathbf{T}\mathbf{x}_i\|^2 - 2(\mathbf{T}^*\mathbf{x}_i)^T(\mathbf{T}^\dagger\mathbf{x}_i - \mathbf{T}\mathbf{x}_i)\\
= {} & \sum_i \|\mathbf{R}^\dagger\mathbf{x}_i + \mathbf{v}^\dagger\|^2 - \|\mathbf{R}\mathbf{x}_i + \mathbf{v}\|^2 - 2(\mathbf{R}^*\mathbf{x}_i + \mathbf{v}^*)^T\\
& (\mathbf{R}^\dagger\mathbf{x}_i + \mathbf{v}^\dagger - \mathbf{R}\mathbf{x}_i - \mathbf{v})\\
= {} & \sum_i \|\mathbf{R}\mathbf{x} + \mathbf{v} + \varepsilon(\mathbf{v}^* - \mathbf{v} + \mathbf{b}^+ \times \mathbf{R}\mathbf{x}_i)\|^2 -\\
& \|\mathbf{R}\mathbf{x}_i + \mathbf{v}\|^2 - 2(\mathbf{R}^*\mathbf{x}_i + \mathbf{v}^*)^T\\
& (\mathbf{R}\mathbf{x}_i + \varepsilon\mathbf{b}^+ \times \mathbf{R}\mathbf{x}_i + \mathbf{v} + \varepsilon(\mathbf{v}^* - \mathbf{v}) - \mathbf{R}\mathbf{x}_i - \mathbf{v})\\
= {} & \sum_i 2\varepsilon\big((\mathbf{v}^* - \mathbf{v} + \mathbf{b}^+ \times \mathbf{R}\mathbf{x}_i)^T(\mathbf{R}\mathbf{x}_i + \mathbf{v}) -\\
& (\mathbf{R}^*\mathbf{x}_i + \mathbf{v}^*)^T(\mathbf{b}^+ \times \mathbf{R}\mathbf{x}_i + \mathbf{v}^* - \mathbf{v})\big) + o(\varepsilon^2) \qquad (15)
\end{aligned}
$$

If ε is small enough, we can discard terms in $o(\varepsilon^2)$.

$$
\begin{aligned}
\Delta\mathbf{E} \simeq {} & \sum_i 2\varepsilon(\mathbf{v}^* - \mathbf{v} + \mathbf{b}^+ \times \mathbf{R}\mathbf{x}_i)^T(\mathbf{R}\mathbf{x}_i - \mathbf{R}^*\mathbf{x}_i + \mathbf{v} - \mathbf{v}^*)\\
= {} & 2\varepsilon\Big(-n\|\mathbf{v}^* - \mathbf{v}\|^2 + (\mathbf{v}^* - \mathbf{v})^T\big(\sum_i \mathbf{R}\mathbf{x}_i - \sum_i \mathbf{R}^*\mathbf{x}_i\big)\\
& + \sum_i (\mathbf{b}^+ \times \mathbf{R}\mathbf{x}_i)^T(\mathbf{R}\mathbf{x}_i - \mathbf{R}^*\mathbf{x}_i)\Big)\\
= {} & 2\varepsilon\Big(-n\|\mathbf{v}^* - \mathbf{v}\|^2 + \sum_i (\mathbf{b}^+ \times \mathbf{R}\mathbf{x}_i)^T(\mathbf{R}\mathbf{x}_i - \mathbf{R}^*\mathbf{x}_i)\Big)\\
= {} & -2\varepsilon\Big(n\|\mathbf{v}^* - \mathbf{v}\|^2 + \sum_i (\mathbf{b}^+ \times \mathbf{R}\mathbf{x}_i)^T\mathbf{R}^*\mathbf{x}_i\Big) \qquad (16)
\end{aligned}
$$

We now show that the sum in (16) is also positive. Using the matrix representation of rotation,

$$
\begin{aligned}
& \sum_i (\mathbf{b}^+ \times \mathbf{R}\mathbf{x}_i)^T\mathbf{R}^*)\\
= {} & \sum_i \Big((z\sum_j \mathbf{R}\mathbf{x}_j \times \mathbf{R}^*\mathbf{x}_j) \times \mathbf{R}\mathbf{x}_i\Big)^T\mathbf{R}^*\mathbf{x}_i\Big)\\
= {} & z\sum_{i,j}\big((\mathbf{R}\mathbf{x}_j \times \mathbf{R}^*\mathbf{x}_j) \times \mathbf{R}\mathbf{x}_i\big)^T\mathbf{R}^*\mathbf{x}_i\\
= {} & z\sum_{i,j}\big(\mathbf{R}^*\mathbf{x}_j(\mathbf{R}\mathbf{x}_j)^T\mathbf{R}\mathbf{x}_i - \mathbf{R}\mathbf{x}_j(\mathbf{R}^*\mathbf{x}_j)^T\mathbf{R}\mathbf{x}_i\big)^T\mathbf{R}^*\mathbf{x}_i\\
= {} & z\sum_{i,j}\mathbf{x}_i^T\mathbf{R}^T\mathbf{R}\mathbf{x}_j\mathbf{x}_j^T(\mathbf{R}^*)^T\mathbf{R}^*\mathbf{x}_i - \mathbf{x}_i^T\mathbf{R}^T\mathbf{R}^*\mathbf{x}_j\mathbf{x}_j^T\mathbf{R}^T\mathbf{R}^*\mathbf{x}_i\\
= {} & zn\sum_i \mathbf{x}_i^T\mathbf{C}\mathbf{x}_i - \mathbf{x}_i^T\mathbf{R}^T\mathbf{R}^*\mathbf{C}\mathbf{R}^T\mathbf{R}^*\mathbf{x}_i > 0 \;\; \forall\mathbf{R} \neq \mathbf{R}^*. \qquad (17)
\end{aligned}
$$

In the last equation $\mathbf{C}$ is the covariance matrix of the $\mathbf{x}_i$. The last inequality is justified by the fact that the rotation matrix $\mathbf{R}^T\mathbf{R}^*$ breaks the alignment between the principal component of $\mathbf{C}$ and the direction

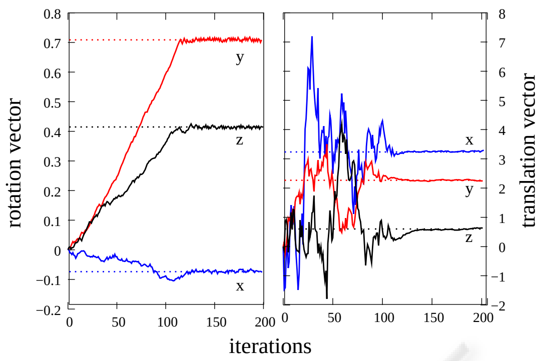

Figure 1: Convergence of the algorithm. The three vector components are indicated by *x*, *y* and *z*. The dotted horizontal lines are the true parameter values of rigid body transformation and the solid lines show the evolution of the estimated values using the learning algorithm.

of maximum variance in $\mathcal{V}$. Putting (17) and (16) together shows that $\mathbf{E}$ decreases when moving from $\mathbf{T}$ to $\mathbf{T}^\dagger$. There is thus no local minima in $\mathbf{E}$, so $\mathbf{E}$ is a Lyapunov function of the system, which proves the convergence.

6 EXPERIMENTS

The first experiment aims at illustrating the convergence properties of the algorithm described above and is performed in simulation. A rotation vector $\mathbf{b}^*$ and a translation vector $\mathbf{v}^*$ were randomly generated. The estimated rigid body transformation was initialized to the identity $\mathbf{b} = \mathbf{v} = \mathbf{0}$ and the algorithm was run on randomly generated points $\mathbf{x}_i$. The results can be seen in Figure 1. One sees that both $\mathbf{b}$ and $\mathbf{v}$ converge to $\mathbf{b}^*$ and $\mathbf{v}^*$ respectively, as is expected from the convergence properties studied above.

The next experiment involves a tracking task in a real stereo vision setting made of two low quality USB cameras mounted on a fixed support. Three color patches were taped on the object to be tracked. A software, based on the OpenCV library can track color blobs and locate them in three dimensions. The object was moved by hand, so the only information about the position of the object is given by the stereo vision system. So the real position of the object is unknown, i.e., there is no ground truth.

Using the data recorded from the stereo vision system, the position and orientation of the end-effector were computed using two different algorithms, the iterative one described in this paper (5) and (6) and the closed-form solution described in (Horn, 1987). This algorithm finds the rigid body transformation by optimizing a least square criterion similar to $\mathbf{E}(\mathbf{T})$ defined

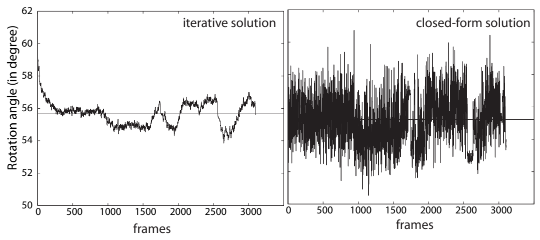

Figure 2: The behaviors of the tested algorithms in case of noise in the data. The iterative algorithm (left graphs) is less noisy than the closed-form algorithm. The object is static, and the same data was used on both algorithms.

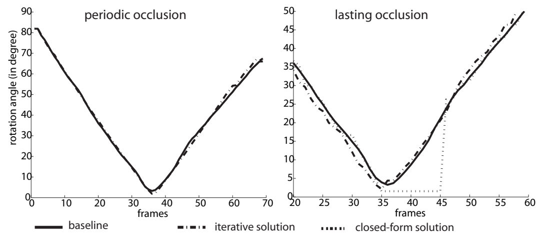

Figure 3: The behaviors of the tested algorithms in case of occlusions. The iterative algorithm can deals well with points periodically missing and points missing for a number of consecutive frames (lasting occulsions).

in (8).
In both cases, the data was taken as it is, without any preprocessing. The iterative algorithm was initialized using the closed-form algorithm on the initial patch positions. In the absence of ground truth, the precision of the tracking algorithm is not investigated. Rather, we compare the behaviors of the iterative and closed-form algorithms

The first experiment was made with a static object. Using the same marker position data coming from the stereo vision software, we ran both algorithms to estimate the position and the orientation of the object. The results can be seen in Figure 2. One sees that the iterative solution is much less sensitive to noise in the data. This is because the closed-form solution has no memory, whereas the iterative solution can only update its current estimate up to a certain amount, which produces a smoothing effect.

The second experiment was made with a moving object. In this experiment, the effect of missing points is investigated. Two different scenarios were tested. In the first scenario (periodic occlusion), a randomly selected point was removed in each frame. In the second scenario (lasting occlusion) a given point was removed from the data for 10 consecutive frames. The results can be seen in Figure 3. One sees that for both scenarios, the closed-form algorithm (dotted lines) cannot deal with the missing points as it requires at least three concomitant points. To the contrary, the iterative algorithm (dashed-dotted line) can deal with the missing points as it has no such requirement. It can follow pretty well the position given by the baseline (solid line). This baseline was obtained by using the closed-form algorithm and smoothing the result.

7 DISCUSSION

The results presented above show that an iterative solution to a rigid body transformation can be advantageous in a tracking application. The main advantages come from the fact that the iterative solution does not make the assumption that it has concomitant points. Moreover, it ensures a continuity in the estimates, which is not guaranteed by the memoryless closed-form solution. When using the iterative solution suggested here, the learning rate must be carefully chosen to be big enough to avoid loosing track of the object, while remaining small enough to ensure a smooth estimate of the transformation.
Although it was not investigated in this paper, we believe that the suggested algorithm could be useful in other applications, especially in iterative algorithms like ICP. This algorithm could also most probably be easily extended to include uniform scaling of rigid body transformations.

REFERENCES

Altmann, S. (1986). *Rotations, Quaternions and Double Groups*, chapter 4, page 80. Oxford University Press.

Arun, K., Huang, T., and Blostein, S. (1987). Least-squares fitting of two 3-d point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Eggert, D., Lorusso, A., and Fisher, R. (1997). Estimating 3-d rigid body transformation: a comparison of four major algorithms. *Machine Vision and Applications*.

Hestenes, D. (1999). *New Foudations for Classical Mechanics*, pages 277–305. Fundamental Theories of Physics. Kluwer Academic Publishers, 2 edition.

Horn, B. (1987). Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4(4):629–641.

Walker, M., Shao, L., and Volz, R. (1991). Estimating 3-d location parameters using dual number quaternions. *CVGIP: Image Understanding*.