Multi-Person Tracking Strategies Based on Voxel Analysis
(Draft)

C. Canton-Ferrer, J. Salvador, J.R. Casas, and M.Pardàs

Technical University of Catalonia, Barcelona, Spain,
{ccanton,jordi,josep,montse}@gps.tsc.upc.es

Abstract. This paper presents two approaches to the problem of simultaneous tracking of several people in low resolution sequences from multiple calibrated cameras. Spatial redundancy is exploited to generate a discrete 3D binary representation of the foreground objects in the scene. Color information obtained from the zenithal view is added to this 3D information. The first tracking approach implements heuristic association rules between blobs labelled according to spatiotemporal connectivity criteria. Association rules are based on a cost function which considers their placement and color histogram. In the second approach, a particle filtering scheme adapted to the incoming 3D discrete data is proposed. A volume likelihood function and a discrete 3D re-sampling procedure are introduced to evaluate and drive particles. Multiple targets are tracked by means of multiple particle filters and interaction among them is modeled through a 3D blocking scheme. Evaluation over the CLEAR 2007 database yields quantitative results assessing the performance of the proposed algorithm for indoor scenarios.

1 Introduction

The current paper addresses the problem of detecting and tracking a group of people present in an indoor scenario in a multiple camera setup. Robust, multi-person tracking systems are employed in a wide range of applications, including SmartRoom environments, surveillance for security, health monitoring, as well as providing location and context features for human-computer interaction.

A number of methods for camera based multi-person 3D tracking have been proposed in the literature [5]. A common goal in these systems is robustness under occlusions created by the multiple objects cluttering the scene when estimating the position of a target. Single camera approaches [8] have been widely employed but are more vulnerable to occlusions, rotation and scale changes of the target. In order to avoid these drawbacks, multi-camera tracking techniques [2] exploit spatial redundancy among different views and provide 3D information as well. Integration of features extracted from multiple cameras has been proposed in terms of image correspondences [3], multi-view histograms [10] or voxel reconstructions [6].

We propose two methods for 3D tracking of multiple people in a multi-camera environment. Redundancy among cameras is exploited to obtain a binary 3D voxel representation of the foreground objects in the scene as the input of the tracking system. The first approach processes the information as follows: a time-consistent label is assigned to each blob corresponding to a person in the room and the 3D position of the person is updated at every frame. All the processing in this step is performed using heuristic criteria such as closest blob, most similar color, etc.

Filtering techniques may also be employed to add temporal consistency to tracks. Kalman filtering approaches have been extensively used to track a single object under Gaussian uncertainty models and linear dynamics [8]. However, these methods do not perform accurately when facing noisy scenes or rapidly maneuvering targets. Particle filtering has been applied to cope with these situations since it can deal with multi-modal *pdf*s and is able to recover from lost tracks [1]. In the second proposed tracking system, a particle filter is implemented to track a target estimating its 3D centroid and no motion model has been assumed to keep a reduced state space. Particle weights are evaluated through a volume likelihood function measuring whether a particle falls inside or outside a volume. A 3D discrete re-sampling technique is introduced to propagate particles and to capture object shifts. Multiple targets are tracked assigning a particle filter to every one. In order to achieve the most independent set of trackers, we consider a 3D blocking method to model interactions. It must be noted that this second tracking system with particle filtering has been already introduced in [4].

Finally, effectiveness of the proposed algorithms is assessed by means of objective metrics defined in the framework the CLEAR07 [7] multi-target tracking database.

2 System Overview

This section aims to describe in a brief manner the main blocks composing our multi-person tracking system. The input data are images captured by five calibrated cameras and their respective calibration data. Four of those cameras are placed at the corners of a meeting room and the fifth is installed as a zenithal camera.

The first block in our system is an adaptive foreground segmentation block based on the Stauffer-Grimson method [12] applied over each of the input images. It consists of two different working phases: the initialization step, when the segmentation algorithm does not know yet the contents of the background and the adaptive loop, when some model of the background has already been acquired but it still needs to be updated to cope with phenomena such as slow ambient light variations.

After this first block, a Shape from Silhouette algorithm applied on the obtained foreground masks delivers a 3D reconstruction of the foreground in the instantaneous scene by testing voxel occupancy over the available foreground

masks coming from each camera. In order to obtain useful foreground volumes, voxel visibility over each camera is used to allow the classification as foreground of voxels out of the common visibility volume for the five cameras.

The third and final block consists in the tracking of 3D connected components, or blobs. This block gets 3D binary foreground volumes from the previous Shape from Silhouette stage and the color image from the zenithal camera to perform a tracking based on both available kinds of information. Two systems are proposed based on this visual information. The first approach process the information as follows: a time-consistent label is assigned to each blob corresponding to a person in the room and the 3D position of the person is updated at every frame. In this approach, a cost function based on heuristic criteria such as closest blob, most similar color, etc. is used to solve the tracking temporal correspondence problem. The second approach employs a particle filtering strategy using only the information coming from the reconstruction (thus not taking into account color information). A particle filter is assigned to each person and connectivity criteria are employed to drive the particles. Finally, an exclusion criteria is employed to separate particles among different filters.

2.1 Foreground Segmentation

During initialization, the method needs to be fed with images containing only background elements, such as a table used during a meeting, whiteboards, chairs in their original position, etc. The initial algorithm estimates both the mean YUV values of every pixel of the background and their variances, assuming a single modal Gaussian behavior for each pixel.

The adaptive loop can also be divided in two phases: firstly it decides whether to classify a pixel in an image as belonging to the background or the foreground of the scene by using the available Gaussian model. The decision of belonging to the foreground will only be taken when the difference in chrominance with respect to the mean in the Gaussian model is higher than a threshold computed from the variance of the latter and the pixel's luminance is below a certain threshold, because of the unreliability of the chrominance vector for high luminance values. The second phase consists in updating the Gaussian model for each pixel classified as background with the new data from the current frame. Thus, the foreground segmentation is able to adapt to slight variations in the light condition of the captured sequence by continuously learning the background.

2.2 Shape from Silhouette

Before the actual tracking step, a 3D model of the foreground of the scene is obtained as the result of a Shape from Silhouette (SfS) algorithm delivering voxelized foreground blobs from a set of foreground segmentation masks of calibrated cameras' images. Such algorithm applies a consistency check on the projection of the elementary spatial analysis unit (a voxel in this case) over the five available cameras to obtain the analysis result for each spatial sample. In this case, the test consists in checking if the analyzed voxel can be considered as part of the

foreground by observing if the content of the foreground masks in the area of projection of the voxel is part of the foreground.

For simplicity, the consistency check of the algorithm only uses the projection of voxels' geometrical centers over the five available cameras. To speed up the execution time, a look-up table (LUT) containing the pixel coordinates of the projection of every voxel center on every camera image is generated using the calibration data previously to the actual analysis. Another LUT, containing the camera visibility for every voxel in a completely empty space is computed for the visibility-aware consistency check.

Finally, a non-linear function is applied to decide whether a voxel is occupied by a foreground element. Using camera visibility in the consistency check, a voxel is considered as part of the foreground if:

1. The voxel is seen by all five cameras and the consistency check over the five cameras is positive
2. The voxel is seen by only four cameras and the consistency check over the four cameras is positive
3. The voxel is seen by only three cameras and the consistency check over the three cameras is positive

Otherwise, the voxel is considered as part of the background. This technique delivers 3D models for the foreground of the scene with enough accuracy, even in areas with low camera visibility, thanks to the visibility LUT. Effects such as occlusion by background elements, i. e. tables or chairs, is not correctly treated by this approach, but as mentioned above the obtained results are enough accurate to our target application.

3 Heuristics Based Tracking

The heuristic tracker receives the binary 3D voxelized blobs from the Shape from Silhouette step as well as the camera images from the zenithal camera. As a first step, an analysis of connected components on the input volume is applied in order to get an identification label for each blob. Spurious foreground voxels are also removed in this first step.

3.1 Blob classification

For each blob, its volume, the position of its centroid and its height are computed to obtain a first classification for the foreground blobs of the scene. This classification takes into account the following rules:

1. If the blob has a volume smaller than a given threshold, it is considered as an object, otherwise it is marked as a person
2. If the blob is marked as a person and it is taller than a certain height (140 cm), the blob is marked as a standing person, otherwise it is marked as a sitting person

3.2 Color model for blobs

In addition to the extracted features mentioned above, a color model of the blob is obtained from the information contained in the image from the zenithal camera. Keeping in mind that our system is designed to be fast enough to deliver results for applications running in real-time, we decided to create a color model using only the color of a layer of voxels of each blob. In addition, we also wanted those layers to be as much populated as possible. This condition took us to the decision of choosing a layer of voxels at a height of 100 cm for sitting persons and 150 cm for standing persons, heights at which the areas of the sections of the body in those gestures present high values. Thus, in the case of blobs classified as a sitting person, the color model is obtained from the projection over the zenithal camera of the centers of the voxels at a height of 150 cm. Similarly the sampling of the sitting person is obtained from the voxels at 100 cm of height.

The color model obtained through the mentioned sampling is a RGB histogram with a parametric number of bins. In our tests, 16 bins delivered the best results.

3.3 Tracking

The tracking algorithm is based on the application of heuristic rules. Once the relevant features (blob classification, color model) have been extracted, a cost function is computed from the available data from tracked blobs in the previous instant and (candidate) blobs in the current time instant.

Firstly, a marginal cost is computed as the 2D euclidean distance between each pair of tracked and candidate blobs. If the distance is shorter than a certain speed times the time difference between frames, assuming that such speed is the maximum speed of a person in a meeting room, this first marginal cost is set to zero. Otherwise, if the distance is longer than the maximum distance but smaller than twice such maximum distance, a cost is set from the formula [distance / mindistance - 1]. If the distance is larger than twice the maximum distance, the cost is set to 1. Thus, the marginal cost for the euclidean distance is set as a value comprised in the range [0, 1]. This extension to the maximum possible speed of a person (up to twice the expected value) is aimed to balance the effects of blobs merging, usually implying a very high speed of blobs' centroids.

A second marginal cost is computed as the 1-complementary of the Bhattacharyya distance computed from the histograms of each pair tracked-candidate, resulting in a value also comprised in the range [0, 1], although in general the dynamic range of this type of distance is much smaller than the whole range.

With those two marginal costs and the remaining information from the blob classification (blobs' volumes), a cost table is generated computing the potential association among any two candidate blobs from two sequential frames. If a tracked blob has only one candidate blob with a distance cost smaller than 1, then the marginal cost for the euclidean distance is used for all the pairs formed with this tracked blob. If there are several candidates with a distance cost smaller than 1 for a given tracked blob, then for those tracked-candidate pairs the color

marginal cost is used instead of the distance cost. Furthermore, if the candidate is classified as an object instead of as a person, a penalty is applied by multiplying by 2 the color cost.

When the cost table is filled in, the candidate blob with less cost is assigned to each tracked. If a candidate is assigned to more than one tracked blob, then a special group ID is added to the label of each of the tracked blobs to allow a correct later separation. Otherwise, if a candidate is only assigned to one tracked blob, the color model for that tracked is updated by averaging its previous histogram with the candidate's one. If a candidate is not assigned to any tracked blob and the blob classification reported it as being a blob from a person, a new tracked is added to the list with copying the data from the feature extraction. Finally, if a candidate does not have any matching candidate (cost smaller than 1), it is removed from the list of tracked blobs.

4 Particle Filtering Applied to 3D Tracking

Particle Filtering (PF) is an approximation technique for estimation problems where the variables involved do not hold Gaussianity uncertainty models and linear dynamics. The current tracking scenario can be tackled by means of this algorithm to estimate the 3D position of a person $\mathbf{x}_t = (x, y, z)_t$ at time t, taking as observation a set of binary voxels representing the 3D scene up to time t denoted as $\mathbf{z}_{1:t}$. Multiple people might be tracked assigning a PF to each target and defining an interaction model to ensure track coherence.

For a given target $\mathbf{x}_t$, PF approximates the posterior density $p(\mathbf{x}_t|\mathbf{z}_{1:t})$ with a sum of N_s Dirac functions:

$$p\left(\mathbf{x}_t|\mathbf{z}_{1:t}\right) \approx \sum_{j=1}^{N_s} w_t^j \delta(\mathbf{x}_t - \mathbf{x}_t^j), \tag{1}$$

where w_t^j are the weights associated to the particles and $\mathbf{x}_t^j$ their positions. For this type of tracking problem, a Sampling Importance Re-sampling (SIR) PF is applied to drive particles across time [1]. Assuming importance density to be equal to the prior density, weight update is recursively computed as:

$$w_t^j \propto w_{t-1}^j \, p(\mathbf{z}_t|\mathbf{x}_t^j). \tag{2}$$

SIR PF avoids the particle degeneracy problem by re-sampling at every time step. In this case, weights are set to $w_{t-1}^j = 1/N_s$, $\forall j$, therefore

$$w_t^j \propto p(\mathbf{z}_t|\mathbf{x}_t^j). \tag{3}$$

Hence, the weights are proportional to the likelihood function that will be computed over the incoming volume $\mathbf{z}_t$ as defined in Sec.4.1. The re-sampling step derives the particles depending on the weights of the previous step, then all the new particles receive a starting weight equal to $1/N_s$ which will be updated by the next volume likelihood function.

Finally, the best state at time t of target m, $\mathbf{X}_t^m$, is derived based on the discrete approximation of Eq.1. The most common solution is the Monte Carlo approximation of the expectation as

$$\mathbf{X}_t^m = \mathbb{E}\left[\mathbf{x}_t|\mathbf{z}_{1:t}\right] \approx \frac{1}{N_s}\sum_{j=1}^{N_s} w_t^j \mathbf{x}_t^i. \tag{4}$$

The major limit of PF, and specially SIR ones, is the capability of the particle set of representing the *pdf* when the sampling density of the state space is low. Scenarios with high number of degrees of freedom require a large number of particles to perform an efficient estimation with the consequent increase in terms of computational cost. An unnecessary computational load could appear with a number of particles larger than required.

Up to authors knowledge, the novelty of the proposed of scheme is to employ the minimum unit of the scene, the voxel, to redefine state space sampling. Being our volume a discrete representation, particles are constrained to occupy a single voxel and move with displacements on the 3D discrete orthogonal grid.

4.1 Likelihood Evaluation

Function $p(\mathbf{z}_t|\mathbf{x}_t)$ can be defined as the likelihood of a particle belonging to the volume corresponding to a person. For a given particle j occupying a voxel, its likelihood may be formulated as

$$p(\mathbf{z}_t|\ \mathbf{x}_t^j) = \frac{1}{|\mathcal{C}(\mathbf{x}_t^j, q)|}\ \sum_{\mathbf{p}\in\mathcal{C}(\mathbf{x}_t^j, q)} d(\mathbf{x}_t^j, \mathbf{p}), \tag{5}$$

where $\mathcal{C}(\cdot)$ stands for the neighborhood over a connectivity q domain on the 3D orthogonal grid and $|\mathcal{C}(\cdot)|$ represents its cardinality. Typically, connectivity in 3D discrete grids can be 6, 14 and 26 and in our research $q = 26$ provided accurate results. Function $d(\cdot)$ measures the distance between a foreground voxel $\mathbf{p}$ in the neighborhood and the particle.

Ideally, particles placed inside the volume of the target achieve maximum likelihood while those being on the surface of the volume attain a non-zero value. Volumes belonging to people would be completely solid but, in practice, there are holes introduced as the effect of segmentation inaccuracies during the SfS reconstruction.

4.2 3D Discrete Re-sampling

The re-sampling step has been defined according to the condition that every particle is assigned to a foreground voxel. In other words, re-sampling has usually been defined as a process where some noise is added to the position of the re-sampled particles according to their weights [1]. The higher the weight, the more replicas will be created. In our current tracking scenario, re-sampling adds

some *discrete* noise to particles only allowing motion within the 3D discrete positions of adjacent foreground voxels as depicted in Fig.1a. Then, non populated foreground voxels are assigned to re-sampled particles. In some cases, there are not enough adjacent foreground voxels to be assigned, then a connectivity search finds closer non-empty voxels to be assigned as shown in Fig.1b.

(a) (b)

Fig. 1. Discrete re-sampling example (in 2D).

No motion model has been assumed in the space state in order to keep a reduced dimensionality of our estimation problem. However, object translations are captured within the re-sampling step by means of this particle set expansion leading to satisfactory results.

4.3 Multi-Person PF Tracking

Challenges in 3D multi-person tracking from volumetric scene reconstruction are basically twofold. First, finding an interaction model in order to avoid missmatches and target merging. The second is filtering spurious objects that appear in scene reconstruction and discarding non-relevant objects such as chairs or furniture. This last problem is managed by the last module of the system that performs a higher semantic analysis of the scene.

Several approaches have been proposed [2, 10] but the joint PF presented in [9] is the optimal solution to multi-target tracking using PFs. However, its computational load increases dramatically with the number of targets to track since every particle estimates the location of all targets in the scene simultaneously. The proposed solution is to use a split PF per person, which requires less computational load at the cost of not being able to solve some complex cross-overs. However, this situation is alleviated by the fact that cross-overs are restricted to the horizontal plane in our scenario (see Fig.2a).

Let us assume that there are M independent PF trackers, being M the number of humans in the room. Nevertheless, they are not fully independent since each PF can consider voxels from other tracked targets in both the likelihood evaluation or the 3D re-sampling step resulting in target merging or identity missmatches. In order to achieve the most independent set of trackers, we consider a blocking method to model interactions. Many blocking proposals can be found in 2D tracking related works [9, 11] and we extend it to our 3D case. Blocking methods penalize particles that overlap zones with other targets. Hence,

(a) (b)

Fig. 2. Particles from the tracker A (yellow ellipsoid) falling into the exclusion zone of tracker B (green ellipsoid) will be penalized by a multiplicative factor $\alpha \in [0, 1]$.

blocking information can be also considered when computing the particle weights as:

$$w_t^j = \frac{1}{N_s}\, p(z_t|x_t^j) \prod_{\substack{k=1 \\ k \neq m}}^{M} \beta\left(X_{t-1}^m, X_{t-1}^M\right), \tag{6}$$

where M is the total number of trackers, m the index of the evaluated tracker and X is the estimated state. Term $\beta(\cdot)$ is the blocking function defining exclusion zones that penalize particles that fall into them. For our particular case, considering that people in the room are always sitting or standing up (this is a meeting room so we assume that they never lay down), a way to define an exclusion region modeling the human body is by using an ellipsoid with fixed x and y axis. Axis in z is a function of the estimated centroid height. An example of this exclusion technique is depicted in Fig.2. Tracked objects that come very close can be successfully tracked even though their volumes have partially merged.

4.4 Parameter tunning

Two parameters drive the performance of the algorithm: the voxel size ν and the number of particles N_P. Experiments carried out explore the influence of these two variables on the *MOTP* and *MOTA* scores as depicted in Fig.3. This plot shows how scenes reconstructed with a large voxel size do not capture well all spatial details and may miss some objects thus decreasing performance of the tracking system. Furthermore, the larger the number of particles the more accurate the performance of the algorithm; however, no substantial improvement is achieved for more than 600 particles due to the restriction imposed that every particle occupies the size of one voxel. Visual results of these effects are depicted in Fig.4.

This experiments allowed setting the two defining parameters of the algorithm for the test phase as follows: $\nu = 3$ and $N_P = 300$.

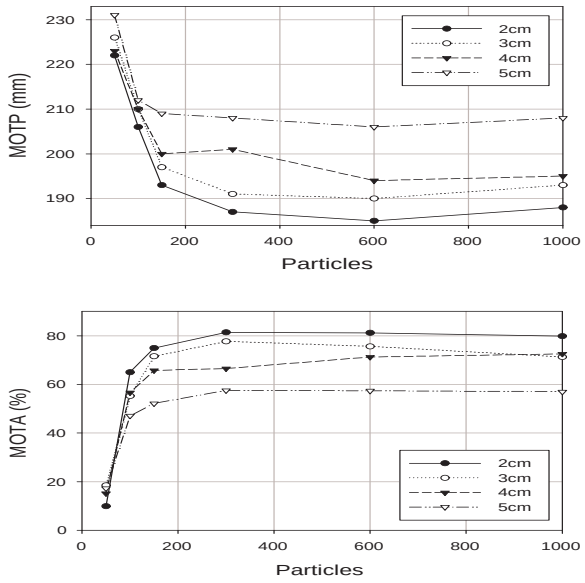


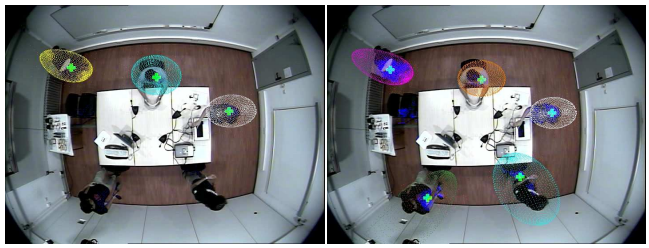
Fig. 3. PF tracking system performance. *MOTP* and *MOTA* scores for various voxels sizes and number of particles. Low MOTP and high MOTA scores are preferred indicating low metric error when estimating multiple target 3D positions and high tracking performance.

5 Results and Conclusions

Results for the two proposed systems are shown in Table 1. The PF tracker obtains a better performance over the heuristic tracking system in terms of misses, false positives and missmatches. This effect is acchieved since the PF keeps information about the multimodal structure of the pdf of the tracked object rendering it more robust to poor observations, even if no color information is taken into account in this case. Obviously, the introduction of color features for the PF tracker is the next target in our future work.

System	MOTP	$\overline{m}$	$\overline{fp}$	$\overline{mme}$	MOTA
Heuristic Tracker	168	27.74%	40.19%	1.58%	30.49%
PF Tracker	147	13.0%	7.78%	0.79%	78.36%

Table 1. Quantitative results with voxel size of 3 cm. Legend: misses ($\overline{m}$), false positives ($\overline{fp}$) and mismatches ($\overline{mme}$).



(a) Experiments with $\nu = 5$cm and $\nu = 2$cm. 300 particles employed.

(b) Experiments with 100 and 300 particles. Voxel size set to $\nu = 2$cm.

Fig. 4. PF tracking system zenital view of two comparative experiments. In (a), two tracking runs showing that large voxel reconstructions miss some objects. In (b), two tracking runs in a scene involving sudden motion showing how a reduced number of particles filter lose track of one target.

References

1. Arulampalam, M.S., Maskell, S., Gordon, N., Clapp, T.: A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. IEEE Trans. on Signal Processing, vol. 50:2, pp. 174–188, 2002.
2. Bernardin, K., Gehrig, T., Stiefelhagen, R.: Multi and Single View Multiperson Tracking for SmartRoom Environments. Proc. CLEAR Workshop, LNCS, vol. 4122, 2006.
3. Canton-Ferrer, C., Casas, J.R., Pardàs, M.: Towards a Bayesian Approach to Robust Finding Correspondences in Multiple View Geometry Environments. LNCS 3515:2, pp. 281–289, 2005.
4. López, A., Canton-Ferrer, C., Casas, J.R.: Multi-Person 3D Tracking with Particle Filters on Voxels. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP), 2007.
5. Checka, N., Wilson, K.W., Siracusa, M.R., Darrell, T.: Multiple person and speaker activity tracking with a particle filter. Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP), vol. 5, pp. 881–884, 2004.
6. Cheung, G.K.M., Kanade, T., Bouguet, J.Y., Holler, M.: A real time system for robust 3D voxel reconstruction of human motions. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), vol. 2, pp. 714–720, 2000.
7. CLEAR Evaluation Workshop, 2006 and 2007.

8. Focken, D., Stiefelhagen, R.: Towards vision-based 3D people tracking in a smart room. IEEE Int. Conf. on Multimodal Interfaces (ICMI), pp. 400–405, 2002.
9. Khan, Z., Balch, T., Dellaert, F.: Efficient particle filter-based tracking of multiple interacting Targets using an MRF-based motion model. Proc. Int. Conf. on Intelligent Robots and Systems, vol. 1, pp. 254–259, 2003.
10. Lanz, O.: Approximate Bayesian Multibody Tracking. IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 28:9, pp. 1436–1449, 2006.
11. MacCormick, J., Blake, A.: A Probabilistic Exclusion Principle for Tracking Multiple Objects. Int. Journal of Computer Vision, vol. 39:1, pp. 57–71, 2000.
12. Stauffer, C., Grimson, W.E.L.: Learning patterns of activity using real-time tracking. IEEE trans. on Pattern Analysis and Machine Intelligence, **22(8)**, August 2000