Automated Depth Video Monitoring For Fall Reduction : A Case Study

Josh Brown Kramer
Ocuvera
jbrownkramer@ocuvera.com

Lucas Sabalka
Ocuvera
lsabalka@ovuvera.com

Ben Rush
Ocuvera
brush@ovuvera.com

Katherine Jones
University of Nebraska Medical Center
kjonesj@unmc.edu

Tegan Nolte
Ocuvera
tnolte@ovuvera.com

Abstract

Patient falls are a common, costly, and serious safety problem in hospitals and healthcare facilities. We have created a system that reduces falls by using computer vision to monitor fall risk patients and alert staff of unsafe behavior before a fall happens. This paper is a companion and followup to "Modeling bed exit likelihood in a camera-based automated video monitoring application," in which we describe the Ocuvera system. [1] Here additional details are provided on that system and its processes. We report clinical results, detail practices used to iterate rapidly and effectively on a massive video database, discuss details of our people tracking algorithms, and discuss the engineering effort required to support the new Azure Kinect depth camera.

1. Introduction

Patient falls are a common, costly, and serious safety problem in hospitals and healthcare facilities. Approximately 2 to 3% of hospitalized patients fall each year [16, 3] resulting in nearly one million falls in U.S. hospitals; approximately one-fourth of these falls result in injury [18, 16] and approximately 11,000 are fatal. [7] The cost of care for the 2% of patients who sustain serious injury from a fall is nearly $14,000 greater than for patients who fall without serious injury. [25] When a patient falls unobserved, hospitals must rule out and/or diagnose and treat injuries by conducting imaging studies and performing procedures. 34% of the 530 inpatients who fell in an academic medical center underwent imaging with an average cost of $900 per study, and 20% of patients imaged had at least one positive finding. [8] Fall-related injuries are designated a preventable Hospital-Acquired Condition by the Centers for Medicare and Medicaid (CMS). [9] CMS does not reimburse hospitals for costs incurred from patient falls. This policy has not significantly decreased injury associated with hospital falls, and there is a lack of evidence for changes in hospital procedures that can significantly reduce falls. [24]

One limitation of current fall risk reduction interventions is the inability to prospectively predict when patients will exit a bed with enough lead time for healthcare professionals to respond and meet the patient's needs. Addressing this limitation presents an opportunity to decrease the incidence of falls and their sequelae, which decrease quality of life and increase health care costs. Over 80% of falls in hospitals may be unobserved, making it difficult to determine patient location at the time of the fall. [17] Unobserved and unassisted falls frequently follow unobserved and unassisted bed exits. In a recent study, 45% of falls originated from the bed, 21% from a bedside chair, 21% occurred in the bathroom after the patient exited either the bed or chair, and 13% originated elsewhere (e.g. hallway, shower, commode). [13] Injury is twice as likely with an unassisted fall as with an assisted fall. [23] Thus, decreasing the incidence of falls and fall-related injuries in hospital rooms could be achieved through decreasing unattended bed exits as a common precursor to falls.

To that end, we at Ocuvera have created a computer vision system that estimates the probability of a bed exit in the near future and sends an alert to nursing staff if that probability rises above a configurable threshold. It consists of a collection of "room clients" (depth cameras attached to computers) that are placed in patient rooms, and a mobile app for monitoring video and receiving and responding to alerts. We use depth for improved computer vision and deidentification. Previously we have given an overview of the system. [1] Particular attention is given there to state estimation, predictive modeling, and theoretical results achieved on video of patients. In this paper, we report clinical results, detail practices we use to iterate rapidly and effectively on a massive video database, discuss details of our people tracking algorithms, and discuss the engineering effort used for the new Azure Kinect depth camera.

2. Related work

While progress has been made recently to reduce falls in hospitals, and there is evidence that multicomponent approaches can reduce falls, [20] their multifactorial causes and the continued prevalence of falls [5] are evidence that additional approaches are still needed. [20, 7] The Ocuvera system can compensate for many of the shortcomings of pressure pads, sitters, and central video monitoring (CVM) systems:

- Pressure pads are common and affordable. However, a recent, large, and well-respected study found that they did not decrease fall rates, possibly due to high false alarm rates and alarm fatigue. [21]
- Use of sitters is costly due to high labor expenses. Studies of reduced sitter usage demonstrated no increase in fall rates, while studies of implementing sitters demonstrated mixed results, [15] possibly due to decreased use of standard interventions when sitters were present. [2]
- CVM is a lower cost alternative to sitters [6] that uses humans to continuously observe video of up to 15 patients from a central location. [6, 19, 11, 4] CVM systems can reduce fall rates by 20-29%. [6, 19, 12] However, the literature documents limitations of CVM, including potential for human error (e.g. attention fatigue) by monitoring technicians, [19] delay in responding to patient behavior as technicians relay information to nursing personnel, [4] ineffective hand-off communication between technicians and nursing personnel, [4, 19] privacy concerns of patients and staff, [11] and the ongoing cost of training technicians. [10]

Other marketed fall risk reduction solutions use a variety of approaches to address some of these limitations. For example, one technology provides audio and video monitoring of patients at risk of falling by transmitting live video from hospital rooms to a central monitoring station where monitoring technicians watch the patient feeds. If the monitoring technician sees a patient begin to leave their bed, a voice command is often issued first to discourage the movement and, if that is not heeded, an alert is issued to nursing staff. Another technology offers a remote patient monitoring solution which pushes live patient video to mobile phones and includes virtual bed rails. When patient movements are detected outside of these superimposed, invisible rails, an alert is sent to provider staff.

Because falls are a complex problem with no single root cause, fall risk reduction interventions are best implemented as a bundle that can address multiple factors contributing to patient falls. When implemented as part of a fall risk reduction bundle, our system contributes a unique combination of real-time, automated, predictive, video-based, 3-dimensional algorithms nurses can use to reduce fall risk.

	Baseline	**Intervention**
Total Admissions	3966	3618
Total Study Patients	193	130
Unassisted Falls	26	11
Assisted Falls	9	9
Total Falls	35	22
Injurious Falls	10	2

Table 1. Raw Numbers For Study Enrollment and Fall-Related Outcomes Among 13 Hospitals

	Baseline	**Intervention**	**Change**
Unassisted Falls	6.56	3.04	-54%
Assisted Falls	2.27	2.49	10%
Total Falls	8.83	5.53	-37%
Injurious Falls	2.52	0.55	-78%

Table 2. Results Per 1000 Admissions

3. Clinical results

In January 2018, we completed USDA NIFA SBIR Phase I project 1012701 to evaluate the feasibility of a prototype version of the system at critical access hospitals. In June 2018, we completed a Nebraska Research and Development (R&D) Phase I grant study that focused on developing an adaptive training program for system usage. These studies included 15 hospitals in baseline and intervention stages. During the baseline stage, when alerts and video were not sent to nursing staff, 4.7% of admissions participated in the study. During the intervention stage when nurses did receive alerts and video, 4.0% of admissions participated in the study. Results of these studies were combined and have been submitted for separate publication. Preliminary results of that analysis includes the following observations. [14]

Nine sites appeared to use the system to intervene to prevent patients at high risk for falls from exiting the bed unassisted resulting in an 89% decrease in unattended bed exits/day (0.84 to 0.09); 96.5% of these patients were at high risk for falls. Five sites appeared to use the system to monitor patients as they exited the bed unassisted resulting in a 639% increase in the rate of unattended bed exits/day (0.43 to 3.18); 89% of these patients were at high risk for falls. It appears that the five monitoring hospitals may have used the system to intervene among patients at high risk for falls and to monitor patients at low risk for falls to ensure they could safely increase their mobility without assistance.

The sensitivity of the system to detect unattended bed exits among all hospitals was 97.4% (calculated from bed exits during the baseline stage). The positive predictive value

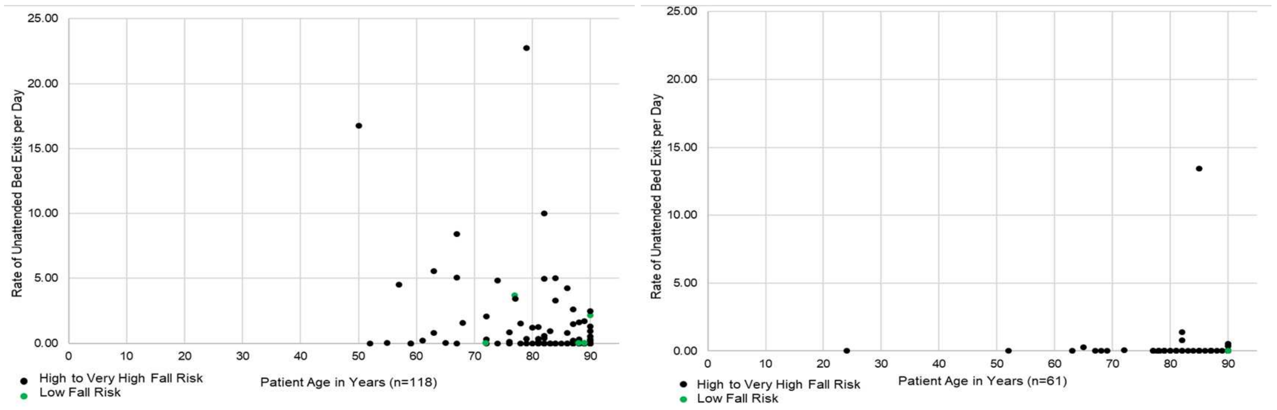

Figure 1. Association between the rate of unattended bed exits rates and patient age during Baseline (left) and Intervention stages (right) at nine study sites using the system as intended, showing use of the Ocuvera system effectively prevented unattended bed exits.

of the system was 59.4%. One or two nurses independently reviewed video associated with the 4,190 alerts sent during the intervention stage of the two studies. Altogether, 2,487 (59.4%) were classified by at least one nurse as true positives because they alerted nurses to behavior that warranted assessment of the patient's needs. Later deployments have seen positive predictive values exceeding 70%. Nurses report the impact of false alerts is less for this system because it is video-based, allowing them to quickly visually assess patient needs.

During the baseline stage, the median lead-time was 28 seconds for 318 unattended bed exits associated with 64 unique patients. Lead-time is the number of seconds that elapsed between the first predictive alert generated by the system due to specific patient movement until a bed exit. We calculated a lead-time for patients who exited the bed unattended during baseline if the patient started lying down in bed and was alone from the time the movement precipitating the bed exit began until three seconds after the bed exit.

Two of the 15 hospitals did not report complete data for admissions or fall event data. In the 13 hospitals included in the final analysis, 4.3% of total admissions participated in the study. From baseline to intervention unassisted falls/1000 admissions decreased by 54% (6.56 to 3.04) and injurious falls/1000 admissions decreased by 78% (2.52 to 0.55). Assisted falls/1000 admissions increased by 10% from baseline to intervention (2.27 to 2.49). An assisted fall occurs when hospital staff lower a patient to the ground in a controlled manner (ideally using a gait belt); assisted falls are significantly less likely to result in injury than unassisted falls. It is possible that assisted falls increased during the intervention phase due to the predictive system alert that allowed nurses to observe patient behavior, make an informed decision, and provided the lead-time to reach the patient and provide assistance including controlling a fall.

Our system has been deployed at a local hospital since 2017 for over 700 patients at high risk for falls and over 6,500 patient days. Unassisted fall rates have shown significant improvement (see Figure 2). The unit sees almost 8,000 patient days per year and has experienced a 54% reduction in unassisted falls for patients using the system compared to pre-intervention.

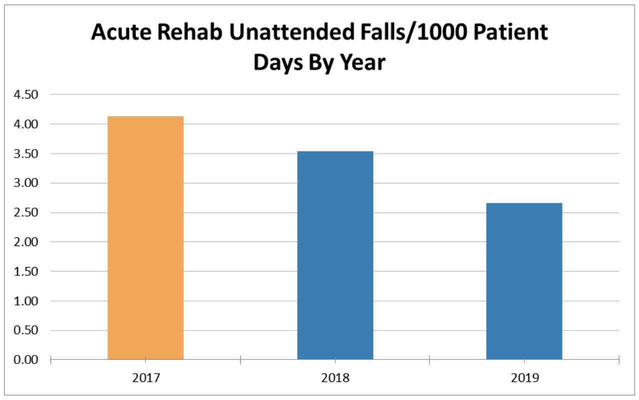

Figure 2. Unattended falls per 1,000 patient days at acute neuro med/surg rehab unit. Our system was introduced gradually in 2017.

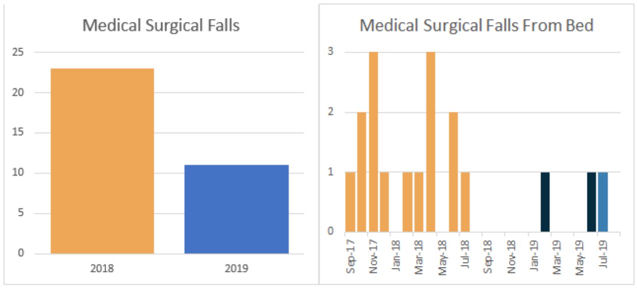

Figure 3. Total falls by year (left) and total falls by month (right). Our system introduced September 2018. The darker bars for the two falls in February and June 2019 occurred with patients whose rooms were not equipped with the system.

At our second clinical deployment, a 36-bed medical surgical unit, intervention was activated September 2018. Falls decreased by 52.2% from 2018 to 2019. At this location, 53% of falls were related to bed exits for 2018. In the 12 months pre-intervention there were 15 falls related to bed exits. In the 12 months post-intervention there were 3, a decrease of 80%. See Figure 3. As of March 2020 there have only been two falls related to bed exits for patients using the system.

Our system was chosen for a trial implementation at a top academic medical center. The pilot implementation involved 16 prototype systems collecting data on over 650 patients. Results indicated a 54% decrease in unassisted falls for patients using the system (an inherently high-fall-risk population) and a 98.4% sensitivity in predicting bed exits.

Across 17 sites combined, with a combined 2473 patients monitored out of over 10,000 patients admitted, unassisted fall rates decreased by an estimated 46%.

4. Test harness

We test algorithmic changes against our database of video. A change is only allowed if it makes an improvement in the end-to-end system. There are several aspects of our testing harness and associated engineering principles that may be of interest, especially for video-based systems that rely on a history of state.

4.1. Metrics

The metrics we use represent the idea that we are preventing bed exits while reducing the number of false alerts. Accordingly, the primary metrics we track when developing our algorithms are **sensitivity**, i.e. the proportion of unattended bed exits that the system alerts on, **positive predictive value (PPV)**, i.e. the proportion of alerts that occur while the patient is engaging in unsafe behaviour, and **number of alerts per patient day**.

Previously, we required an improvement in both sensitivity and PPV before allowing a change. If one went up and one went down, we would have soul-searching philisophical discussions about whether the change was good. More recently, we have adopted a single metric : a weighted sum of sensitivity and number of alerts per day that heavily weights sensitivity. Having a single metric has made a significant improvement in our iteration speed.

4.2. Our unit of indepence is the patient

We want to make sure that we are not reporting metrics on data that we trained against. To us, training data includes any video that was used to make decisions about whether or not to include a change as part of the system. This includes the input to a machine learner, but also the video we worked against while developing hand-written algorithms.

If we were to use video from the same patient that was used as training data, we could expect some frames to be very similar to data the system was trained against. For that reason, we split the patients into test and training patients. We take pains to avoid the R&D team ever seeing test patient data.

This decision impacts how we view the reliability of metrics. In particular, we compute confidence by bootstrapping: suppose the set of training patients has size n. We repeatedly choose a sample of size n with replacement and compute the desired metric. We then pick an interval containing, for instance, 95% of the resulting data. Patients vary wildly in the number and manner of bed exits, so this process has a large effect on the confidence interval. For example, a recent test run on a subset of our data yielded a confidence interval for sensitivity of 92.9% ± 5%. This was computed on 4677 bed exits. If the bed exits had been treated as independent, the estimate would have been ± 1%.

Splitting by patient is not the only method of data division. For example, once our library contains a sufficient number of hospitals, we might split the hospitals into training and test groups. As is, our metrics prove that the system generalizes to new patients. Splitting by hospital will enable us to better prove that our system generalizes to new hospital environments.

4.3. Subsampling

We run new algorithms against a cluster of over 50 commodity machines. Given that we have over 200,000 hours of video, we subsample video rather than running all of it on a cluster run. Bed exits are sparse, so subsampling by itself would yield few bed exits and a very noisy estimate of sensitivity. Instead, our taggers identify nearly all of the bed exits by watching the video at high speed, and we test against all of them. When computing statistics on video for a particular patient, we assign weight 1 to frames of video that were found by humans, and a high weight to the remaining randomly chosen frames, calculated so that the total weight of all frames run through the cluster on a patient equals the number of frames for that patient. This produces an unbiased estimate of our metrics while allowing us to focus on events of interest.

This process, for example, allowed us to accurately estimate that our initial manual tagging process had missed hundreds of bed exits on the basis that we had a handful of high weight bed exits in our tests. It also provided us with a more realistic look at our initial estimates of PPV. Most of the clips we were running on were of bed exits, so by number, almost all of our alerts were true positives. By weight however, our PPV was initially below 50%.

4.4. Discretization of state estimates

Many of the underyling state estimates we track (e.g. is there another person in the room, is the patient in the bed, is their leg hanging off the bed) are most naturally represented

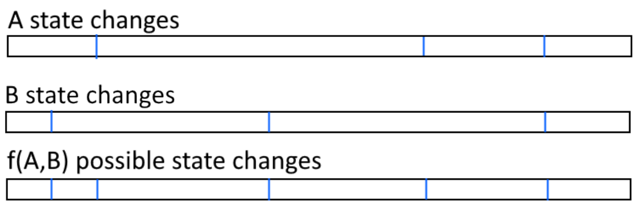

Figure 4. The frames of state changes of $f(A, B)$ are a subset of the frames of state changes of A or B.

as a probability. However, we gain many advantages by thresholding and storing booleans. In particular, we need only store the frames at which the state changes instead of a float per frame. With hundreds of millions of frames per test run and dozens of state estimates, this makes a substantial difference in storage space and retrieval times.

Another advantage of this data structure is that we can quickly compute a function of two or more states in time proportional to the number of state changes instead of frames. This is true because the set of state change frames for this function must be a subset of the state change frames for the originals (see Figure 4.4). For example, suppose we want to know how often we estimate that a particular patient is sitting up in bed when no one else is in the room over the course of a particular day. If the patient sat up 20 times and someone entered and left the room 24 times, the total computation takes on the order of 44 operations, in contrast to the half a million frames that this computation represents.

This opens up the ability to quickly run experiments without re-running the computer vision. For example, we are able to experiment with different alerting policies, different sensitivity levels, and different test metrics, among others. In fact, our alert policy is a decision tree that is a function of the history of discretized states. For a given tree, we are able to quickly compute a timeline of the current node in the tree. This allows us to choose from among hundreds of candidate splitting rules at each node, and thereby build a tree that operates on history of state.

4.5. Reproducibility

There are two ways we want our system to be reproducible: we want the system to have the same output from test run to test run, and we want the behavior in the field to be the same when that video is run in the lab.

Reproducibility from test run to test run allows us to differentiate between changes in metrics that were due to algorithmic changes and changes due to random fluctuations. There are two main ways that non-determinism sneaks into our algorithms. First, some of our algorithms rely on randomness. For example, we use RANSAC to find the floor. The solution we have settled on is to always use a random seed that is a function of the incoming frame. Another source of non-determinism is parallelism. If a parallel algorithm depends, even subtly, on the order in which computations finish, that is a source of non-determinism. For example, our bed finding iterates over many candidate beds in parallel. If there are ties in fitness, this can affect the determinism of the algorithm. We have handled issues like this on a case-by-case basis. In this case, we assign an index deterministically to each bed and break ties by index.

We also want video of an event to run in the lab the same as it did in the field. The major impedement we face is that the behavior of the system depends on a history of video. The way we have resolved this is to reset the system if there has been 30 seconds of no motion in the scene. Thus, if we want to know how much lead time our alerts would provide for a particular bed exit, we back up the video to the most recent time there were 30 seconds without motion and run the video through our algorithms from there.

It should be noted that we perform this reset in the field. There is an argument that in practice, access to more state history should be better for the system. While that may be true, it may also be false, and we have no ability to test it. Instead, we know through testing on the cluster that the system performs well with resets after periods without motion. This is not merely theoretical. For example, because our bed finding is expensive, we previously kept our bed model even after 30 seconds with no motion. It turned out that, very rarely, the system would find a bad bed that it thought was good. This model would often be stuck in this state for extensive periods of time, including after several days of use. The testing system gave us no visibility into this problem, since it would only run on short clips, where bad beds were very rare. Now that the bed is recomputed after 30 seconds of no motion, we have not had this problem.

5. People tracking

A key component of our system is logic around identifying and tracking movements of patients, nurses, and other people in the scene. As with many tasks, we believe the future is deep learning, and we are actively developing those algorithms now. However, we are a small company with limited resources. Patient privacy is a fundamental concern, so video-based data of patients in hospitals is very difficult to obtain: it took us 3 years to collect any data, and 2 more years to collect the 200,000 hours we have now. Falls are also very rare events, typically occurring around 4 times every thousand patient-days. Thus, to accelerate time to market, and due to team bandwith limitations, budgetary limitations in data acquisition, and the paucity of data, our current PeopleTracking algorithms take a more classical approach.

The PeopleTracking pipeline processes each depth frame that comes in sequentially, performing all operations in approximately 10ms per frame. The pipeline consists of a number of steps (see Figure 5).

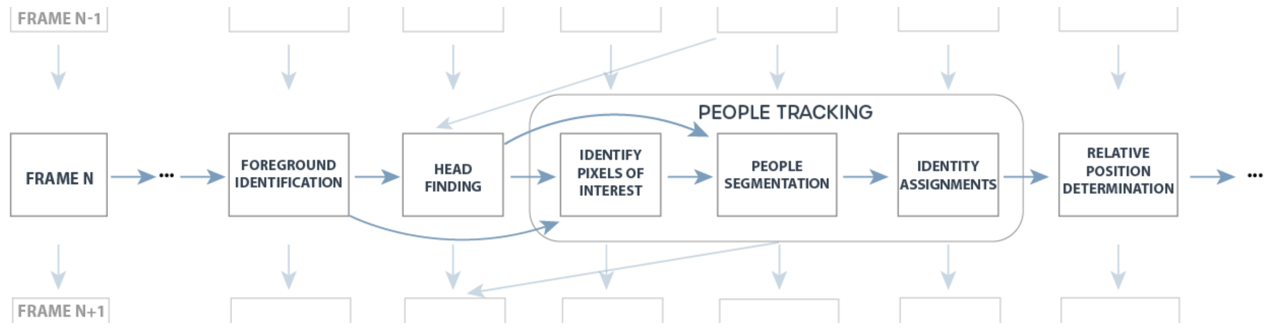

Figure 5. The basic flow of information in our people algorithm.

5.1. Foreground identification

We begin by isolating pixels of interest. One main reason for a pixel to gain interest is recent motion. Each pixel in the image is modeled using a probability distribution based on its recent mean and variance. Pixels which fall outside their expected distribution are considered in motion. Depth-based mathematical morphology operations are then applied to filter out noise and in-fill moving segments.

5.2. Head identification

Another main signal for a pixel to be of interest is if it represents a point on a person's head. We use Microsoft's per-pixel decision tree algorithm, [22] trained on our depth data, to classify pixels as part of a head or not. To speed up the computation, not all pixels are tested. Tested pixels are selected from a subgrid on the image. Foreground pixels in the subgrid are tested, as are pixels from previously identified heads, and also a random selection of background pixels. Then, for positive identifications, a second pass is conducted to fill in classifications for all nearby pixels. Finally, pixels are segmented using depth-aware segmentation. Resulting sufficiently large segments are called heads.

5.3. Pixels of interest identification

Pixels of interest are amalgamated into a single mask: foreground pixels, pixels on identified heads, and pixels physically near identified heads (representing potential shoulders and upper torso). Aside from the head, body parts are not explicitly identified (as with skeletonization algorithms) because identification is unnecessary and because many poses seen are not easily learned due to occlusions from and proximity to nearby objects (other people, tray tables, bedding, bed rails, IV stands, etc.). The mask has various morpholology operations applied to it to filter out small objects like cups and some blanket movement from areas that could represent a person.

5.4. Segmentation and identity assignment

All pixels of interests are then segmented using depth-aware segmentation. Each segment is assigned a class number. Segments that have substantial overlap and physical proximity to segments from a recent previous frame are assigned the same class number. Significant conditional statements determine how to handle conflicts between class assignment. For instance, a conflict can occur when a nurse assists a patient, so from a segmentation perspective the pixels are connected by an isthmus of pixels (the nurse's arm). As another example, a conflict can happen when a person sets an object down. A final example of a conflict is when a patient starts moving in bed and moves their head and feet before moving their torso. In this situation, the head and feet can be tracked separately as moving segments, but they should be combined when it is clear they are connected by the torso. With conflicts resolved, we assign which segments are people versus objects and which segment represents the patient. In general, when handling conflicts between segments, we err on the side of caution and make decisions to ensure we are correctly tracking the patient when they are attempting to exit the bed, as alerting in this situation is the highest priority of the system.

5.5. Relative position determinations

The system can designate exactly one tracked person as the patient. Once that designation is made, patient status is inherited across frames as above. To initially determine which segment represents the patient, segments are evaluated based on their current position relative to the bed and where that segment was first tracked (i.e. in bed, in the frame but out of bed, or from out of frame). Segments can be first seen in the bed or in the frame if the system is turned on while the patient is in the bed or following a system reset after a period of no motion (see Section 4.5).

The system also keeps track not just of where a segment came from, but how confident it is that the segment is a person, where the head is, and how confident it is that it has detected the person's head for the segment, whether the person is near the bed or leaning over the bed, whether the segment has ever had a head detected, when the segment last moved, what its estimated center of mass is, etc. This information is used to not only determine the patient, but also decide when the system should not send an alert because a nurse is in the room helping the patient: if the patient exits the bed with help from a nurse, the system should not disturb nurses with an alert for that bed exit.

6. Azure Kinect

We have adapted the system to use the Azure Kinect depth camera from Microsoft. This section discusses our reasons and the process for integrating with our system.

6.1. Reasons for adopting the Azure Kinect

Availability. We previously used Kinect V2, which is no longer being manufactered.

Quality. The quality and accuracy of the Azure Kinect depth camera is significantly better than other commodity depth cameras on the market.

Power consumption. The camera draws up to 5.9W, compared to 30W for the Kinect V2. This allows us to package our system with a smaller power supply.

Wide field of view. The Azure Kinect has a fisheye lens with a 120°x120° field of view, compared to 70.6° x 60° from Kinect V2. This opens more opportunities for placement, and allows us to see more of the room.

6.2. Goals

Our initial goal in integrating the Azure Kinect was ensuring that it could be used as a drop-in replacement for the Kinect V2. We have a large body of depth video that we want to leverage as we move to this new camera. We want to be sure that quality is not compromised and that test results will apply to the real world. There are several differences between the two cameras that we had to address.

6.3. Wide field of view vs. narrow fields of view

The Azure Kinect operates in two modes: narrow field of view mode, and wide field of view mode. The narrow field of view mode has higher depth quality, but the shape of the IR projector is octagonal, so the corners are clipped (see Figure 6.2). We simulated the effect of clipping the corners on our existing video, and it had a significant negative effect on all of our key metrics. Given this and the fact that we want to eventually use the extra information provided to us by the wide field of view, we have chosen to use wide field of view in production.

6.4. Modifying wide field of view

The wide field of view mode also has differences from the Kinect V2. Most notably, the image is taken with a fisheye lens, and is therefore very distorted. Additionally, the threshold for filtering bad depth data is stronger. We used standard image unwarping in openCV to create a depth image similar to that of the Kinect V2. There were two interesting challenges. First, there was a Moiré pattern in

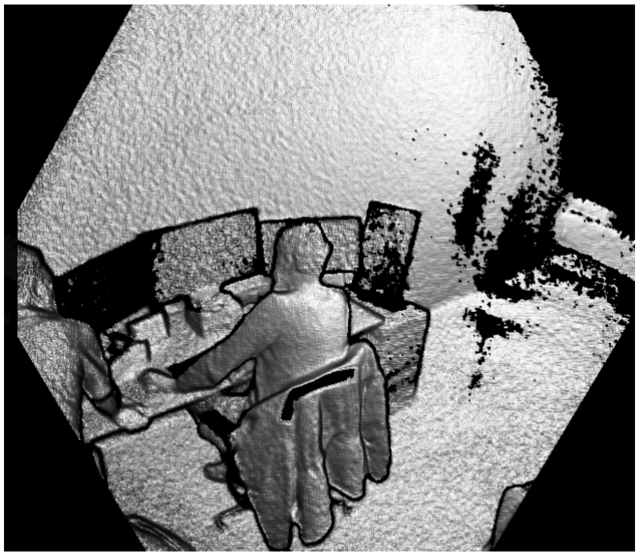

Figure 6. The corners are clipped in narrow field of view mode.

the depth data when we used nearest-neighbor interpolation. This is mostly fixed by using bilinear interpolation, except that zero pixels should not be averaged over in order to avoid smearing artifacts near edges. Eliminating zeros from the average can be achieved efficiently by unwarping the depth image, unwarping a 0-1 valued mask with the same method, and then dividing by this unwarped mask. See Figure 6.4. For the heavy filtering of depth data, we worked with Microsoft to adjust thresholds to get depth data out to the same distances available on Kinect V2. This yields some highly incorrect noise. We filter this by removing small isolated segments (see Figure 6.4). In addition to these challenges, the default 512x512 image from Azure Kinect has a heavy spatial filter. We worked with Microsoft to reduce this filtering effect to produce data closer to the Kinect V2 (see Figure 6.4).

6.5. Testing

To our eye, we are able to produce a signal very similar to that produced by the Kinect V2. To test more systematically, we have run the Azure Kinect system passively on real hospital patients. We have caught some differences this way: namely our motion detection was more likely to trigger a false positive. Having worked through that and other issues, we have now run hundreds of patient days against the Azure Kinect and are satisfied with the results. A direction for the future is learning a map from Kinect V2 style video to Azure Kinect style video for testing purposes. We have good initial results. However, there are challenges. Namely, the CycleGAN [26] model we have learned does not preserve depth effectively, and doing style transfer on every frame of all of our video would be time intensive.

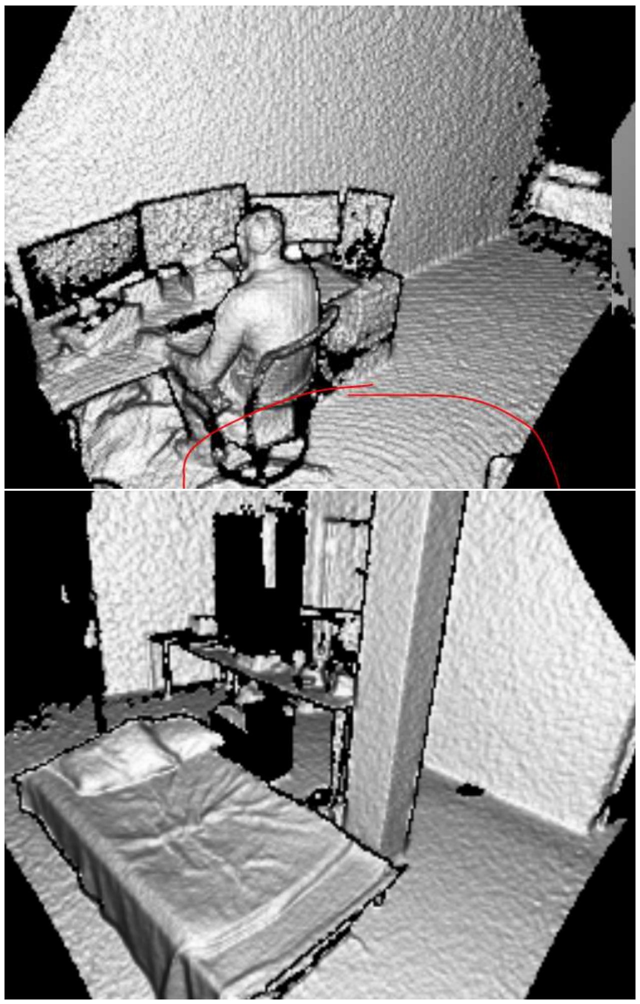

Figure 7. Top: a naive undistortion. The rings are a Moiré pattern. Bottom: bilinear interpolation with proper zero handling. Notice the lack of rings.

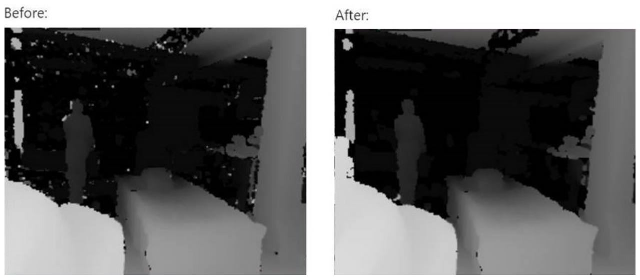

Figure 8. Before: custom permisive noise filter. After: same filter plus small segment removal.

7. Conclusion

In the last few years we have made significant progress in improving the Ocuvera system's performance and support

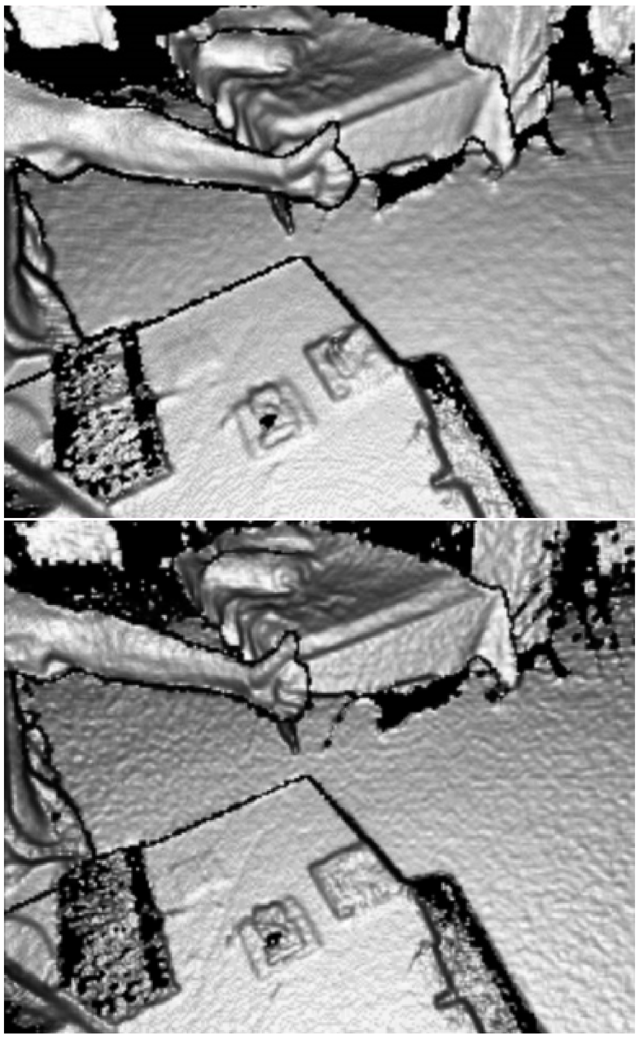

Figure 9. Top: default spatial filtering in 512x512 mode. Bottom: custom spatial filtering more closely matches Kinect V2.

of the Azure Kinect depth camera. These changes amount to a system that can be tested more efficiently, that can track patients more reliably, and that integrates with an updated, improved depth camera. We are constantly working to ensure that our system is both implementing state-of-the-art computer vision techniques and operating as effectively as possible in the field. We believe our results thusfar show a promising technology that can have a positive impact on the complex problem of patient falls in hospitals.

References

[1] P. Bauer, J. Brown Kramer, B. Rush, and L. Sabalka. Modeling bed exit likelihood in a camera-based automated video monitoring application. In *2017 IEEE International Conference on Electro Information Technology (EIT)*, pages 56–61, 2017. 1

[2] D. Boswell, J. Ramsey, M. Smith, and B. Wagers. The cost-effectiveness of a patient-sitter program in an acute care hospital: A test of the impact of sitters on the incidence of falls and patient satisfaction. *Quality Management in Healthcare*, 10(1):10–16, 2001. 2

[3] E.D. Bouldin, E.M. Andresen, N.E. Dunton, M. Simon, T.M. Waters, M. Liu, M.J. Daniels, L.C. Mion, and R.I. Shorr. Falls among adult patients hospitalized in the united states: prevalence and trends. *Journal of Patient Safety*, 9(1):13–17, 2013. 1

[4] K. Bradley. Remote video monitoring: A novel approach in fall prevention. *Journal of Continuing Education in Nursing*, 9(11):29–35, 2016. 2

[5] The Joint Commission. Preventing falls and fall-related injuries in health care facilities, 2015. 2

[6] M. Cournan, B. Fusco-Gessick, and L.Wright. Improving patient safety through video monitoring. *Rehabilitation Nursing*, 43(2):111–115, 2018. 2

[7] L. Currie. *Fall and Injury Prevention. In: Patient Safety and Quality: An Evidence-Based Handbook for Nurses*, chapter 10. Agency for Healthcare Research and Quality, Rockville, MD, April 2008. 1, 2

[8] J. Fields, T. Alturkistani, N. Kumar, A. Kanuri, D. Salem, S. Munn, and D. Blazey-Martin. Prevalence and cost of imaging in inpatient falls: the rising cost of falling. *ClinicoEconomics and Outcomes Research*, 7:281–286, 2015. 1

[9] Centers for Medicare and Medicaid Services. Hospital-acquired conditions, August 2015. 1

[10] D. Goodlett, C. Robinson, P. Carson, and L. Landry. Focusing on video surveillance to reduce falls. *Nursing*, 29(2):20–21, 2009. 2

[11] S. Hardin, J. Dienemann, P. Rudsill, and K. Mills. Inpatient fall prevention: Use of in-room webcams. *Journal of Patient Safety*, 9(1):29–35, 2013. 2

[12] S. Jeffers, P. Searcey, K. Boyle, C. Herring, K. Lester, H. Goetz-Smith, and P. Nelson. Centralized video monitoring for patient safety: a denver health lean journey. *Nursing Economics*, 31(6):298–306, 2013. 2

[13] K.J. Jones. Origin of falls. CAPTURE falls project, 2017. 1

[14] K.J. Jones, L. Sabalka, and G. Hayntazki. Evaluation of automated video monitoring to decrease the risk of unattended bed exits in small rural hospitals. *Journal of Patient Safety*, 2020. in submission. 2

[15] C.E. Lang. Do sitters prevent falls? a review of the literature. *Journal of Gerontological Nursing*, pages 24–33, 2014. 2

[16] J. Mahoney. Immobility and falls. *Clinics in Geriatric Medicine*, 14(4):699–726, 1999. 1

[17] J. Morse. *Preventing Patient Falls*. Sage Publications, Thousand Oaks CA, 2 edition, 2000. 1

[18] D. Oliver, F. Healey, and T. Haines. Preventing falls and fall-related injuries in hospitals. *Clinical Geriatric Medicine*, 26:645–692, 2010. 1

[19] K. Sand-Jecklin, J. Johnson, and S. Tylka. Protecting patient safety: Can video monitoring prevent falls in high-risk patient populations? *Journal or Nursing Care Quality*, 31(2):131–138, 2016. 2

[20] P. Shekelle, R. Wachter, P. Pronovost, K. Schoelles, K. McDonald, S. Dy, and K. Shojania. Making health care safer ii: an updated critical analysis of the evidence of patient safety practices. *Evid Rep Technol Assess (Full Rep)*, 211(1):1–945, 2013. 2

[21] R.I. Shorr, M.A. Chandler, L.C. Mion, T.M. Waters, M. Liu, M.J. Daniels, L.A. Kessler, and S.T. Miller. Effects on an intervention to increase bed alarm use to prevent falls in hospitalized patients: a cluster randomized trial. *Annals of Internal Medicine*, 157(10):692–699, 2012. 2

[22] Jamie Shotton, Andrew Fitzgibbon, Mat Cook, Toby Sharp, Mark Finocchio, Richard Moore, Alex Kipman, and Andrew Blake. Real-time human pose recognition in parts from single depth images. In *CVPR 2011*, pages 1297–1304, 2011. 6

[23] D. Venema, A. Skinner, R. Nailon, D. Conley, R. High, and K. J. Jones. Patient and system factors associated with unassisted and injurious falls: An observational study. *BMC Geriatrics*, 9(1):348, 2019. 1

[24] T.M. Waters, M.J. Daniels, G.J. Bazzoli, E. Perencevich, N. Dunton, V.S. Staggs, C. Potter, N. Fareed, M. Liu, and R.I. Shorr. Effect of medicare's nonpayment for hospital-acquired conditions: Lessons for future policy. *JAMA International Medicine*, pages E1–E8, 2015. 1

[25] C. Wong, A. Recktenwald, M. Jones, B. Waterman, M. Bollini, and W. Dunagan. The cost of serious fall-related injuries at three midwestern hospitals. *Joint Commission Journal on Quality and Patient Safety*, 37(2):81–87, 2011. 1

[26] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks, 2017. 7